NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ANALYSIS AND PROTOTYPING OF THE UNITED STATES MARINE CORPS TOTAL FORCE ADMINISTRATION SYSTEM (TFAS), ECHELON II – A WEB ENABLED DATABASE FOR THE SMALL UNIT LEADER

by

Steven A. Simmons

September 2002

Thesis Advisor: Daniel R. Dolk

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 2002	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE: Analysis and Prototyping of The United States Marine Corps Total Force Administration System (TFAS), Echelon II – A Web Enabled Database for The Small Unit Leader

5. FUNDING NUMBERS

6. AUTHOR(S)
Simmons, Steven A.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT *(maximum 200 words)* This thesis analyzes the requirements and development of a prototype web site for The United States Marine Corps' Total Force Administration System (TFAS), Echelon II – A Web Enabled Database for The Small Unit Leader. The analysis consisted of researching the characteristics of the current manpower system, MCTFS, and the conceptual tenets of the TFAS program. The thesis presents relevant background information on MCTFS, TFAS, and then gives a detailed presentation of system and user functional requirements for TFAS, Echelon II. A detailed system architecture for the TFAS, Echelon II web site prototype and its integration with MCTFS are presented. Specifications are enumerated for a multi-tiered web-enabled application integrated with a system of distributed synchronized databases. Additionally, sample programming code for implemented web site functionality is explained, in conjunction with screen shots of the working prototype. The examples include user identification and data query binding at run-time. Samples of read-only reports and data entry tasks whose data is confined to Marines within the TFAS user's unit are explained. Programming is shown for data entry tasks along with a presentation of how this data is transformed into XML "Unit Diary" files, which are generated dynamically. A migration path is described for the integration of the current web technologies with legacy manpower systems. Finally, conclusions and recommendations derived from the thesis work are presented to aid TFAS planners and developers in moving TFAS from a concept to an operational system.

14. SUBJECT TERMS TFAS, Total Force Administration System, MCTFS, Marine Corps Total Force System, Web-enabled, Relational Database, Distributed Databases, Manpower Systems, XML, Extensible Markup Language, Multi-tier Web Application.

15. NUMBER OF PAGES
144

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

ANALYSIS AND PROTOTYPING OF THE UNITED STATES MARINE CORPS TOTAL FORCE ADMINISTRATION SYSTEM (TFAS), ECHELON II – A WEB ENABLED DATABASE FOR THE SMALL UNIT LEADER

Steven A. Simmons
Major, United States Marine Corps
B.S. Civil Engineering, The Ohio State University, 1990
M.S. Engineering Management, University of Missouri – Rolla, June 1998

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2002

Author: Steven A. Simmons

Approved by: Dr. Daniel Dolk
Thesis Advisor

Dr. Thomas Otani
Second Reader

LCDR Christopher Eagle, USN
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis analyzes the requirements and development of a prototype web site for The United States Marine Corps' Total Force Administration System (TFAS), Echelon II – A Web Enabled Database for The Small Unit Leader. The analysis consisted of researching the characteristics of the current manpower system, MCTFS, and the conceptual tenets of the TFAS program. The thesis presents relevant background information on MCTFS, TFAS, and then gives a detailed presentation of system and user functional requirements for TFAS, Echelon II. A detailed system architecture for the TFAS, Echelon II web site prototype and its integration with MCTFS are presented. Specifications are enumerated for a multi-tiered web-enabled application integrated with a system of distributed synchronized databases. Additionally, sample programming code for implemented web site functionality is explained, in conjunction with screen shots of the working prototype. The examples include user identification and data query binding at run-time. Samples of read-only reports and data entry tasks whose data is confined to Marines within the TFAS user's unit are explained. Programming is shown for data entry tasks along with a presentation of how this data is transformed into XML "Unit Diary" files, which are generated dynamically. A migration path is described for the integration of the current web technologies with legacy manpower systems. Finally, conclusions and recommendations derived from the thesis work are presented to aid TFAS planners and developers in moving TFAS from a concept to an operational system.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENT

I would like to thank my wife, Laura, for all her patience and support during the writing of this thesis and throughout our time at the Naval Postgraduate School. Her love, devotion, and proofreading skills enabled me to complete this challenging thesis work. I would also like to thank Dr. Daniel Dolk and Dr. Thomas Otani for all their instruction and guidance, both in the classroom and in conjunction with this thesis. The research and programming efforts conducted for this thesis would not have been possible without the support and interest of several Marine Corps agencies and their representatives. I would like to thank Major Wadye Smith at Marine Corps Systems Command for assistance in TFAS program research and for providing funding for computer equipment for NPS. Technical assistance was provided by Captain Doug Nelson, Major Edward Reyelts, Mr. Gary Deardorff, and Mr. Clint Swett at the Technology Services Organization (TSO), Defense Finance and Accounting Service (DFAS) in Kansas City, MO. TSO's sponsorship of a research trip to DFAS proved invaluable in development of the TFAS, Echelon II prototype. Lastly, I would like to thank LCDR Chris Eagle, Computer Science Instructor and Interim Department Chair at The Naval Postgraduate School for his talented instruction and inspiration to pursue excellence in Computer Science.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. PURPOSE

The purpose of this thesis is to analyze the requirements and develop a prototype web site for The United States Marine Corps' Total Force Administration System (TFAS), Echelon II – A Web Enabled Database for The Small Unit Leader. This work will evaluate the technical feasibility of developing a web-enabled database by which the small unit leader (company commander and other company level leaders) in the United States Marine Corps can view manpower data on the Marines in the unit and make various non-pay related data updates/changes.

B. AREA OF RESEARCH

The area of research for this thesis deals with multi-tiered web enabled databases, the synchronization of distributed databases, and the integration of web technologies with legacy, proprietary USMC manpower data systems utilizing Extensible Markup Language (XML). Currently, the United States Marine Corps (USMC) is in the planning stages of developing a "Web-Interface" whereby leaders at all levels can view and update data on the Marines in their units over the internet. This effort is called the Total Force Administration System or TFAS. When fully developed, TFAS will replace the current legacy manpower data systems. This thesis and the supporting research will strive to develop the requirements and a working prototype web site for the leaders at the small unit level. This "slice" of the TFAS program for the small unit leader has been designated as Echelon II.

C. BACKGROUND

Currently, the small unit leader has no direct access to manpower data for their personnel and no direct ability to update non-pay related data. At present, Marine leaders must submit requests for data and/or data updates through several layers of command and/or administrators. Requests are eventually processed by specially trained clerks (Unit Diary Clerk, MOS 0151) who interact with a legacy system called Unit

Diary/Marine Integrated Personnel System (UDMIPS) which is the primary system used for the retrieval and updating of USMC manpower data. The UDMIPS software is a heavy client and is one component in several layers of software that ultimately connects to the single source of manpower data resident on a mainframe computer physically resident in St. Louis, MO. This store of USMC manpower data on the mainframe and the related software used to access it is termed The Marine Corps Total Force System or MCTFS. MCTFS is used by the Marine Corps to manage all manpower and pay related data on all active duty, reserve, and retired USMC personnel.

The underlying single source of data resident on the main-frame (MCTFS) is basically a flat file system and is integrally tied to a set of complex COBOL-language programs used to ensure the data is compliant with the hundreds of overlapping manpower and pay policies. For this reason and budgetary constraints, the Marine Corps does not envision restructuring the mainframe system architecture in the near future. However, with the advent of sophisticated web technologies over the past decade, there is no reason the Marine Corps cannot develop a web interface to access MCTFS data and make it widely available to Marine leaders. The software, including UDMIPS, used to access MCTFS data has evolved over the past thirty years into a complex system that few can access, much less use efficiently. UDMIPS as the primary interface to MCTFS data no longer meets the manpower data needs of the leaders of Marines especially in the face of web technologies currently available in COTS (Commercial-Off-the-Shelf) software. The justification for this statement stems from the following deficiencies in the UDMIPS software and the system by which it is employed.

- The UDMIPS software is a heavy client program that must be individually installed on each diary clerk desktop. Updates to this software are frequent (every six months) and it takes quite an effort to update all copies.

- The UDMIPS program is complex and takes weeks of training to gain minimum proficiency. Although the program has become more "Windows-like" over the years, it is not an easy program to master and is not intuitive.

- Data updates via UDMIPS require the user to look up and confirm current "Transaction" codes for every single data item entry. Erroneous transaction codes and incomplete data fields are often the source of either bad data in MCTFS or failed data updates.

- The UDMIPS software is designed for use in the administrative unit in that a unit diary clerk has global visibility to all data and can create data updates to all data items. The safeguard to this current system is that all data updates (transactions) must be verified and electronically signed by the Personnel Officer. The second-order effect of this system is that UDMIPS cannot be distributed to small unit leaders since access to a single "small unit" cannot be set up in UDMIPS.

- Data updates are manpower intensive in that the each data request/data update must be physically passed between many individuals.

- The process of retrieving data and/or making data updates is unresponsive to leader's need for data. Requests for data can take days before they are fulfilled, and data updates can take weeks. The cause for this poor responsiveness is not necessarily due to the legacy software, but is due to the fact that so few people know how to operate UDMIPS and access to the system is closely guarded.

- As of 2000, the manning quantity of unit diary clerks (0151s) was cut by over 1000 and the Battalion/Squadron Administrative Sections were consolidated into Personnel Administration Centers. These PACs now service multiple Battalion sized units. As a result, there are less "diary clerks" to operate the UDMIPs software, whereas the workload has remained constant.

- The second-order effect of the consolidation of the administrative unit is that the "operators" of UDMIPS and their direct management no longer work directly for the "Commander" of the Battalion/Squadron. The Commander has a direct interest in ensuring data in MCTFS is accurate because it affects almost every aspect of the Commander's unit such as promotions, duty assignments, deployments, awards, meritorious boards, etc. Since the administrators no longer work directly for the Commander, they have a less focused interest in maintaining quality data.

- The source of the majority of data updates is typically resident at the Platoon, Company, and Battalion level. The commanders at these levels have no direct access to the MCTFS data and have no direct ability to update the data in a timely manner.

- The diary clerks typically have hundreds of data updates (diary transactions) to process each day. This type of workload on an individual who has no personnel, direct knowledge of the data and the Marines to which it pertains, results in poor data entry and quality control.

D. RESEARCH QUESTIONS

1. Main Research Question

How can a local, multi-tiered, web-enabled database be designed, developed and integrated with current USMC manpower data systems such that the small unit leader may conduct data transactions, while keeping the local data synchronized with the legacy USMC manpower database, the Marine Corps Total Force System?

2. Subsidiary Questions

- What are the functional requirements for TFAS, Echelon II?
- What is an appropriate design for the architecture of TFAS, Echelon II?
- What is an appropriate database schema for the USMC manpower data to support the functional requirements of TFAS, Echelon II?
- What security plan must be devised as part of the design of the TFAS, Echelon II web site to ensure data secrecy and integrity?
- What synchronization strategies can be used to integrate the local backend database for the TFAS, Echelon II web site with the legacy mainframe based USMC manpower data system?

E. SCOPE AND METHODOLOGY

1. Scope

The scope includes:

- Description of existing technical architecture and legacy systems for the USMC manpower data systems.
- General Description of the TFAS Technical Architecture.
- Description of the TFAS Echelon II Web Site and backend database technical architecture.
- Definition and description of the functional requirements of the Echelon II TFAS Web Site for the small unit leader (Company Commander and other appointed company personnel). These requirements will only pertain to "atomic" transactions. Atomic transactions are functions that are ones initiated by the small unit (Company) only and may be processed as a diary immediately without further administrative action by other personnel at other echelons.

- Technical description of the ASP scripts written to implement the functional requirements.

- Description of the integration of the TFAS Echelon II Web Site architecture with existing USMC manpower data systems.

- Description of a proposed general administration of the web site and local database. Including:

  - Define users and roles at small unit level.
  - User account administration.
  - Web site and local backend database administration and maintenance.

- Development of a prototype web site that utilizes a local relational database, which is a replicated version of the Operational Data Store Enterprise (ODSE), kept in St. Louis, MO.

- Demonstrate an operational web site on a server at NPS. The following items will be the technical products of my thesis work:

  - Set up backend database (SQL Sever 2000) containing a copy of the USMC manpower data file.

  - Set up a web server (IIS-5) and load Echelon II HTML and ASP files.

  - Simulate a network of users (Windows 2000 Server Domain) from several units, similar to that found on a USMC base.

  - Set up user accounts for 2 or 3 different companies (small unit level) from different battalions.

  - Demonstrate User authentication.

  - The prototype will demonstrate several different READ Only pages (data display).

  - The prototype will demonstrate several different WRITE pages (data update). The prototype web site will not implement all of the functional requirements defined for TFAS – Echelon II. The thrust of the prototype is to demonstrate that it can work, not to program 50-100 ASP web pages.

  - Demonstrate the web server production of an XML document ready for import into UDMIPS. This XML Document is the data update made by a

small unit leader and is in the XML schema format defined by TSO for the generic import utility of UDMIPS.

- The XML document produced must adhere to the XML Schema defined for UDMIPS.

2. Methodology

The methodology used in this thesis research follows:

- Conduct literature research on current USMC manpower systems.
- Conduct literature research on TFAS program.
- Review sample documents for functional requirements of TFAS - Echelon I (Individual Marine) as template for methodology and structure for development of functional requirements of TFAS Echelon II.
- Conduct telephonic interviews with various USMC agencies responsible for USMC present and proposed manpower data systems. This includes the following major agencies:

  Marine Corps Systems Command
  Manpower & Reserve Affairs (M&RA)
  DFAS/MISSO/TSO Kansas City, MO

- Conduct TAD trip to DFAS, Kansas City to obtain technical advice from TSO, the agency responsible for maintaining the current USMC manpower data systems and ultimately could develop and maintain all echelons of TFAS.
- Obtain training set database of Marine manpower data and convert to SQL Server 2000.
- Obtain XML Schema and samples of the "Unit Diary" XML Document that can be imported into UDMIPS.
- Obtain UDMIPS software.
- Obtain Schema of ODSE.
- Obtain ODSE (Oracle 8i) client software for connecting to the ODSE for database synchronization.
- Obtain current set of Unit Diary Transaction Codes (TTCs) and Prompts required to construct a valid diary. Convert to a relational database so it can

be accessed at run time on the web server to construct the XML Diary document.

- Conduct review of ASP database technologies.

- Conduct review of IIS-5 web server technology.

- Conduct review of Microsoft SQL Server 2000 technology.

- Conduct review of Windows 2000 Server network administration.

3. Assumptions and Limitations

a. Assumptions

- Network Architecture and Software. The United States Marine Corps currently uses Microsoft software for its network operating system software for its intranet/networks aboard bases, stations, and independent units. Specifically, intranets aboard bases and stations are predominantly a hybrid of Windows NT 4.0 and Windows 2000 Server/Advanced Server domains.

- Client Software. Virtually all of the desktop computers within Marine Corps units have a Windows-based operating system. Specially, Windows NT 4.0 Client or Windows 2000 Professional (Client).

- Web Server Software. Because the vast majority of the intranet networks run a Windows Server Operating system (NT 4.0 or Windows 2000 Server), the predominate Web Server being used at the base/station level is IIS-4 or IIS-5 (Internet Information Server).

- Database. Beyond Microsoft Access available in the Microsoft Office (97/2000/XP), there is no widely utilized DBMS (Database Management System) within the Marine Corps. Access is widely used at the local unit level from a single client to a limited client/server role over a single intranet. Access is an adequate DBMS client/server product for limited functions, but is not appropriate as a backend database for larger scale requirements with greater security needs. The requirements for the backend database for the TFAS, Echelon II demand a commercial DBMS. As such, I have selected Microsoft SQL Server 2000 mainly for the ease of integration with the Microsoft based networks and Web servers used throughout the Marine Corps.

b. Limitations

- Atomic Transactions Only. The TFAS, Echelon II Web Site prototype developed for this thesis will only demonstrate "atomic" transactions. Atomic

transactions are those data transactions for updates where the small unit leader has the authority to input the data and submit the diary directly for processing. There are many administrative manpower data tasks in the Marine Corps that require several echelons of commander's/unit's input. These non-atomic or multiple steps manpower data tasks cannot be implemented in the TFAS, Echelon II Web Site prototype since the other echelons of TFAS do not yet exist. No testing of a "Data Cycle." A complete "cycle" of a data update cannot be demonstrated. A complete cycle of a data update would access to actual "live" Marine Corps data systems. As an independent NPS student, I have not been given access to these systems. A description of this process will be given in this thesis. This description could then be implemented by the appropriate Marine Corps Manpower Data Systems agencies.

- Security. Security features of the TFAS, Echelon II Web Site prototype will be addressed in chapter VI. However, the thrust of this thesis and the prototype is a proof of technical concept. Before any actual deployment of the prototype, it would need to be thoroughly analyzed by security experts to ensure that the manpower data being accessed is indeed secure.

- Scale. The TFAS, Echelon II Web Site prototype developed for this thesis will not address issues related to scale. Any actual deployment of the TFAS, Echelon II Web Site prototype could entail a sizable load (number of connected users) on the web and database servers. The TFAS, Echelon II Web Site prototype is being developed on my home computer that has neither the hardware nor software to handle or test heavily web/database traffic. Professional web and database administrators would need to be employed to test the TFAS, Echelon II Web Site prototype.

- Reliability. Reliability is on the other side of the coin of scale. Again, it is beyond the scope of this thesis to analyze and test the reliability of web and database server with a heavy load. Commercial servers and their software have features that provide for fail-over mechanisms and mirror sites both for the web server and database server.

In order to fulfill the purpose of this thesis, the material presented will be organized in the following manner. Chapter 2 will cover background material regarding the current manpower data systems in the Marine Corps. Chapter 3 will present the information about the TFAS program, it's broad conceptual architecture, and functional requirements. Chapter 4 will cover the functional requirements of the TFAS, Echelon II web site prototype developed for this thesis. Chapter 5 will address the system architecture of the prototype. In chapter 6 a description of the programming of the web

pages and database queries necessary to support the functional requirements of the prototype is provided. Finally, Chapter 7 will present recommendations, conclusions, and further work on the TFAS, Echelon II web site.

THIS PAGE INTENTIONALLY LEFT BLANK

II. THE MARINE CORPS TOTAL FORCE SYSTEM

In order to analyze the requirements and develop a prototype for the TFAS, Echelon II web site, an understanding of the Marine Corps' current manpower data systems is required. Such an analysis is necessary because any TFAS web site must interact with the current system, the Marine Corps Total Force System (MCTFS).

A. GENERAL DESCRIPTION

MCTFS "is the single, integrated, personnel and pay system supporting both active and reserve components of the Marine Corps. MCTFS is jointly sponsored/owned by the Marine Corps and the Defense Finance and Accounting Service (DFAS). MCTFS maintains more than 500,000 active, reserve and retiree records that are available to be processed for pay purposes, personnel management or for the production of management reports." [Ref. 1] Each Marine represented in MCTFS has over 1800 data fields that describe the Marine and their status in the Marine Corps. [Ref. 1;p. 1-11] MCTFS uses a single logical database to process transactions at one central location at the Defense Enterprise Computing Center (DECC) located in St. Louis, MO. MCTFS encompasses the personnel and pay data and all computer programs necessary to make changes to the data. The data component of MCTFS is called the Central Master File (CMF), which is in a flat file format. The computer programs associated with MCTFS are written in COBOL and contain over 3 million lines of code. [Ref. 2] These programs are used to make changes to the data and enforce numerous manpower and pay policies and business rules. These programs and the data are tightly integrated so that any change in a particular data field(s) may trigger a cascading effect of logical checks to update other fields or ensure all data required for the transaction were valid and present. This system was first created in 1963 [Ref. 2] and has evolved over the past forty years into a highly complex system which serves as the backbone for dozens of subsidiary manpower and pay data systems.

From a non-technical perspective, The Director, Manpower Management Information Systems Division {CMC (MI)} is responsible for and serves as the program

manager (PM) for the functional management of the manpower (data) portion of MCTFS. [Ref. 1] The technical (computer programming and daily operations) responsibility for the MCTFS mainframe is the responsibility of the Technology Services Organization (TSO), which is a sub-agency of the Defense Finance, and Accounting Service (DFAS) located in Kansas City, MO. Any major modifications to MCTFS must be developed conceptually and funded through The Marine Corps Systems Command (MARCORSYSCOM), the agency responsible for all Marine Corps acquisitions and procurement. These three agencies bear the joint responsibility for operation and maintenance of MCTFS. For example, changes in manpower policy by CMC (MI) would require a modification in the MCTFS data and/or programs. TSO, funded by MARCORSYSCOM, designs (writes the program) and implements the changes on the MCTFS mainframe. Because of the dynamic nature of Marine Corps manpower policy, MCTFS updates are made twice a year in October and May. [Ref. 2] These changes could be reflected in the nature of the data fields (data type, size, new fields, etc.) and/or the programming to enforce a new policy (e.g. automatically promote all E-2s to E-3s who have 18 months time-in-service and have no punitive flags in the system). These policy changes not only affect the data and MCTFS programming, but also have a ripple effect in that they ultimately affect all software systems that are layered on top of MCTFS.

B. MANPOWER DATA PROCESS IN MCTFS

The single source of data for personnel and pay data for the entire Marine Corps, MCTFS, is the backbone through which a variety of software systems interface to access manpower data. To detail all of the various subsidiary systems that "feed" on MCTFS is beyond the scope of this thesis. Rather, we focus on how Marines at the battalion/squadron (units with 400-1000 personnel) interact with MCTFS. All Marines at the battalion level interact with MCTFS indirectly via specially trained personnel located in their administrative unit. Prior to 2000, each battalion/squadron had its own "admin" section staffed by a Warrant Officer, a few Staff Non-Commissioned Officers and some quantity (10-30 depending on the size of the unit) of junior enlisted "admin" clerks (MOS

0151). These admin sections were under the direct command of the battalion/squadron commander and were responsible for administrative functions for that unit. During 2000, a reorganization of the administrative occupational field (01XX MOS) was conducted. The result of this reorganization was the reduction of the manning level by over 1000 administrative personnel. [Ref. 3] Additionally, all of the administrative sections at the battalion/squadron level were consolidated such that a series of "Consolidated Administration" (CONAD) units were formed. These CONADs now serve several battalion-sized units depending on the geography of the base and the location of units served.

The functions served by the administrative units are many and varied. These functions may be grouped into several categories (a few examples are given):

- Individual Marine
  - View Individual Training Record
  - Request an Allotment (automatic pay deduction)
  - Check-in to unit
- Small Unit Leader (Squad/Section, Platoon, or Company)
  - Unit Recall Roster report
  - Physical Fitness Score Report
  - Submit semiannual Proficiency and Conduct marks (performance evaluations) on junior enlisted
- Battalion/Squadron
  - Pay deduction for a Marine who has been punished
  - Promotions
  - Awards

Ultimately, Marine Corps-wide, these administrative units serving the battalions and squadrons conduct about 15 million data transactions per year. [Ref. 3] The total manning of these administrative units is about 2000 people. [Ref. 2] That means, on average, each administrative clerk is processing about 7500 transactions per year or about 100-150 per day (taking into account weekends, holidays, annual leave, and other Marine training).

As with any database, these functions may be grouped into two primary functions READ (get read-only data from MCTFS for a report) and WRITE (make updates to existing data fields, insert new data, or delete data). Currently, all Marines at all levels of command must request these data services though their chain of command. These

requests, typically hard copy in form, eventually are delivered to the administrative unit where the clerks process the request.

C. UNIT DIARY/MARINE INTEGRATED PERSONNEL SYSTEM

The means by which the administrative clerk processes the manpower data requests is through one of two MCTFS application sub-systems: The On-Line Diary System (OLDS) and Unit Diary/Marine Integrated Personnel System (UDMIPS). The OLDS system is the older of the two MCTFS interfaces and was the primary means for interfacing MCTFS for years. The OLDS interface is a black-white "Atari" screen-like application and requires on-line connectivity with a central server. Data input and command prompts are similar to the command-line interface of MS-DOS or UNIX and all data is inputted through a series of alphanumeric codes for transactions and data. The process of making data updates to MCTFS or, rather, the set of data representing the data update, is termed a "unit diary." A unit diary consists of these codes and has very little "readable" data. It is these code-laden unit diary files that the MCTFS mainframe understands and expects when it receives a batch of unit diaries transmitted via the OLDS system. OLDS is extremely difficult to use and for this reason, UDMIPS was developed as its replacement.

Presently, UDMIPS is the primary interface in most administrative units. [Ref. 4] UDMIPS is a heavy client, Windows-driven application. UDMIPS does not require on-line connectivity in order for the clerk to make data updates. Work can be completed off-line and uploaded later when connectivity is re-established. [Ref. 1] Its menus and buttons make entering and retrieving data more intuitive and, thus, much easier to use than OLDS. However, UDMIPS is a jack-of-all-trades in that it contains all functions required at every level of command. There are literally thousands of possible transactions. This makes UDMIPS a very complex system to master. Even though UDMIPS is Windows-driven (menus, etc.), the user must still have knowledge of a great deal of the alphanumeric codes for transactions and data. Every unit diary clerk keeps two 3-inch binders of codes next to their desks for reference as they enter data. These references are MCO P1080.40C Marine Corps Total Force System Personnel Reporting

Instruction Manual (PRIM) and MCO P1080.20M Marine Corps Total Force Systems Codes Manual. Figure 1 below is a sample screen shots of the UDMIPS interface for entering a single Physical Fitness Test Score on an individual Marine:

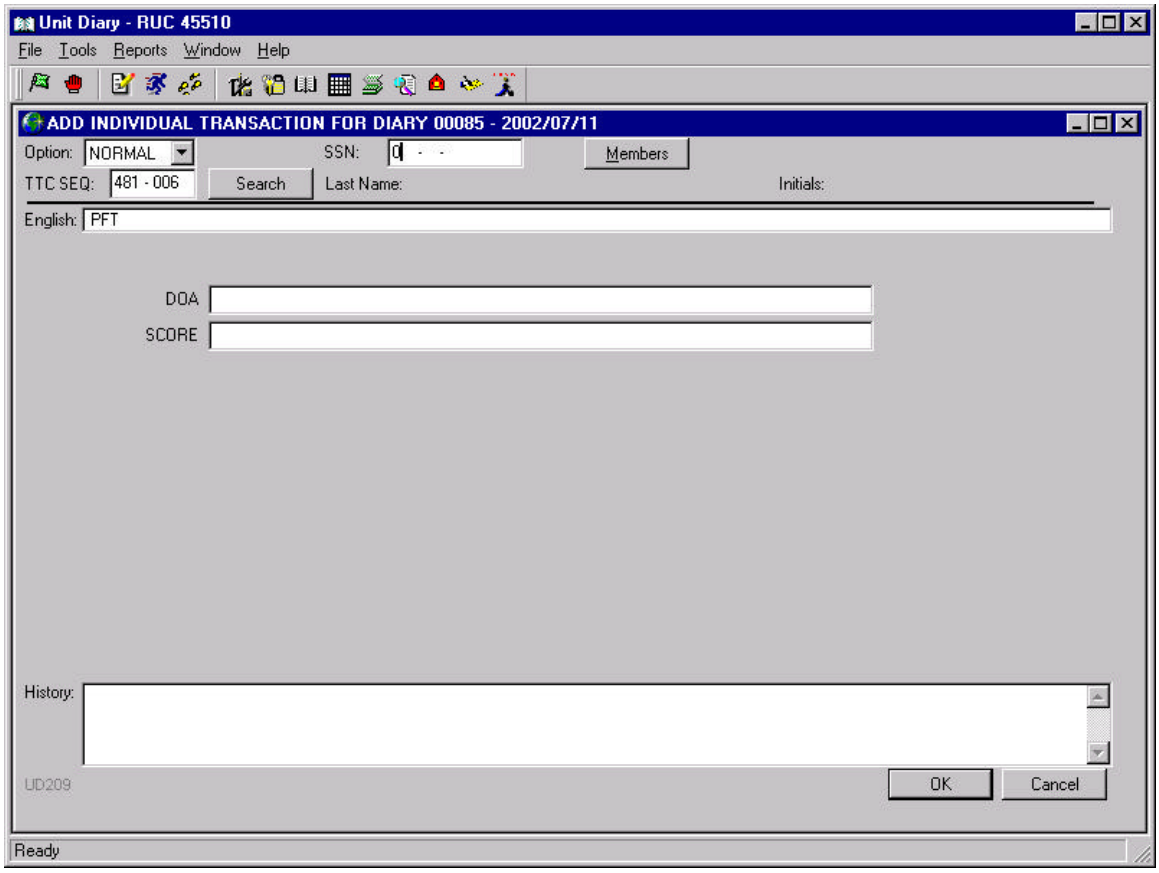


Figure 1. UDMIPS User Interface – Single Diary Transaction

Figure 2 below is a sample screen shot of multiple unit diary transactions for a unit diary.

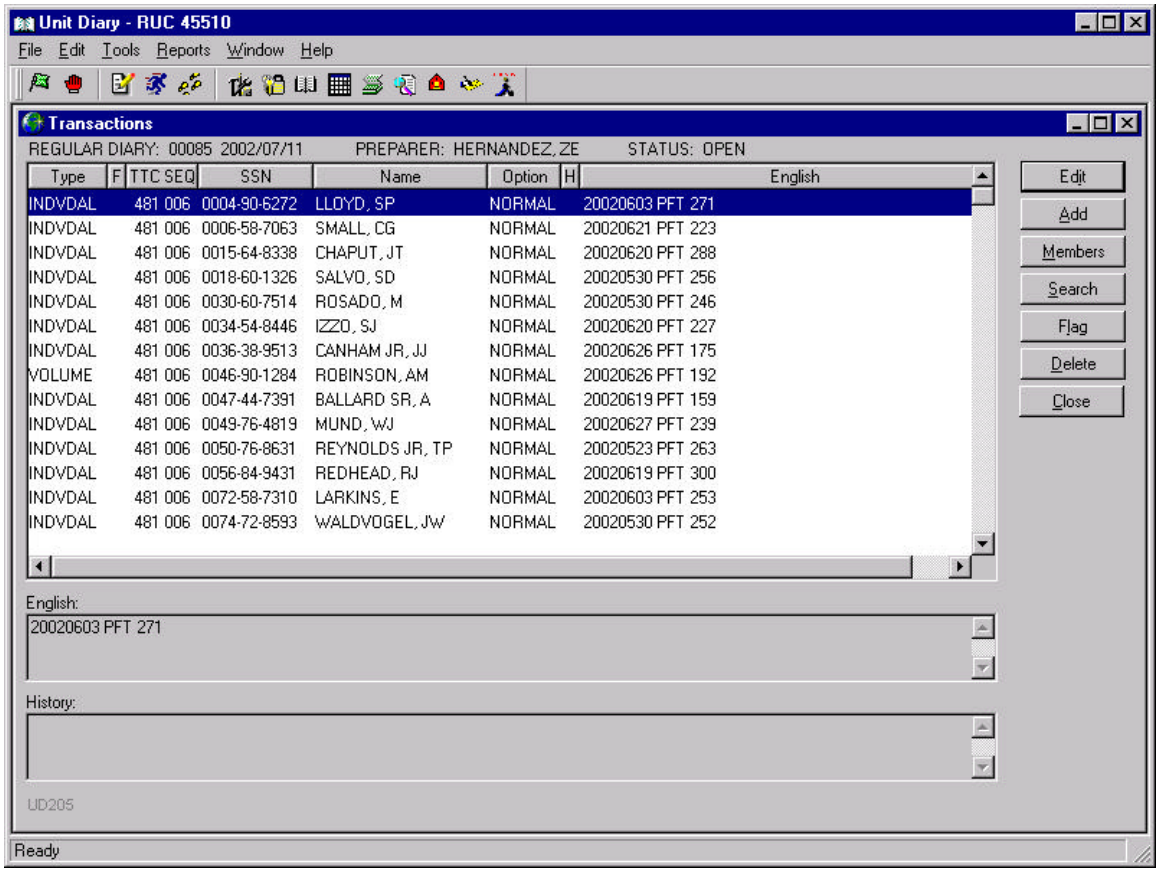


Figure 2. UDMIPS User Interface – Multiple Diary Transactions

1. The Unit Diary

Data updates to the MCTFS data are made by submitting specially formatted files called unit diaries to be processed on the MCTFS main-frame. Both the OLDS and UDMIPS applications perform this task. Because there are thousands of different data transactions (diaries) that could be performed against the 1800 data fields on each of the 500,000 individual Marine records, the level of complexity is obviously immense. However, it is important to understand the basic layout of a typical diary. The following table contains a description of the data items that all diaries must contain regardless of the type of data transaction.

Diary Data Item	Description/Example
System ID	Unique Identifier for Computer System generating the Diary. "A06"
System Code	Unique Identifier for Admin Unit/Diary Preparer "MCSFBN CPL Walters 5640357"
Unit Diary Number	Code generated by the Admin Unit. "Unit Diary 069-02"
Date of Diary	Date the diary was submitted. "20020715"
RUC	Reporting Unit Code. A unique identifier for the battalion-sized unit to which the subject Marine belongs. "11400"
Name of Diary Preparer	"Simmons SA"
SSN of Diary Preparer	"0111223333" Contains a leading 0.
Name of Certifying Officer	"Smart IM"
SSN of Certifying Officer	"0999447777"
Diary Type	Unit Diaries can be one of these: 1) Record of Event 2) Exclusive Entry 3) Individual Entry 4) Group Entry 5) Situational Reporting 6) Volume Transaction
Individual Diary Transaction Data	A single Unit Diary file can have one or more diary transactions (data update to the record of an individual Marine).
Type Transaction Code	A 3-digit numeric code for the diary type of diary transaction. "481" is for a Physical Fitness Test entry on a single Marine. This field is per diary transaction.
Sequence Number	A 3-digit numeric code the subtype of transaction. "006" is a normal PFT score for a marine who took the PFT in the specified semi-annual period and passed. This field is per diary transaction.
Last Name of Marine	This field is per diary transaction. "Jones"
Initials of Marine	This field is per diary transaction. "BO"
SSN of Marine	This field is per diary transaction. "0555116666"
Remarks	There can be one-to-many "remarks" for an individual diary transaction. These are the data for the transaction. For a normal PFT entry, there would be two remarks, the PFT Score and date. "20020303" "261"

Table 1. Unit Dairy Data Fields [After: Ref. 1]

For example, to enter a Physical Fitness Test (PFT) score, the unit diary clerk must have knowledge that there are five different PFT transactions as designated by a MCTFS transaction code called a Type Transaction Code (TTC) and Sequence Code. Together the form the "TTCSEQ" code used by the MCTFS main frame to process a type of transaction. For a PFT data entry task, they are:

TTCSEQ	Description
481 006	*PFT - Marine took scheduled PFT and passed*
481 007	*MED - Marine did not take PFT- Medical Excuse*
481 009	*RNT - Marine required to take PFT, but did not*
481 010	*PAR - Marine took a partial PFT*
481 011	*FAIL - Marine took PFT but failed*

A sample unit diary file with multiple unit diary transactions might look like:

```
A06
MCSFBN 5640357
UNIT DIARY 069-02
20020715
11400
SIMMONS SA
0111223333
6
BUTCHER     TM   0999881234   003000   CERTIFYING OFFICER 0999447777
STEWART     CR   0123456789   335000   20020701 MARKS PRO 4.4 CON 4.5 OCC SC
TROMBA      C    0456784321   481006   20020303 PFT 286
STEVENSON J      3453672233   930314   TO SK 1030 ILL 19980314 HOSPITAL
                                       HIST:  HEART ATTACK
SIMMONS     SA   0718923791   481011   FAILED PFT
```

Figure 3. Sample Unit Diary

This example is not exactly the format of an actual unit diary file, but rather it is presented as a human readable example of the information a unit diary file might contain. Even in this form, it is rather cryptic.

2. The Unit Diary Data Cycle

As unit diary clerks use the UDMIPS application, they obviously need access to the manpower data for reports and as a context for data entry. This data is not provided directly by the MCTFS mainframe. The UDMIPS application running on each admin

clerk's desktop is connected to a local copy of the manpower data resident in MCTFS. This "local database" called the Commanding Officer's Unit Diary Database (CUDDB) is a replicated version of the CMF resident in MCTFS. Ii is also in a flat file, proprietary format. UDMIPS uses this database to read information out to the UDMIPS application. During data entry (the creation of a unit diary), data updates are not written back to this local copy. As stated previously, these data updates are transformed into unit diary files in a format that will be understood by the MCTFS mainframe. A unit diary file may consist of a single data transaction or could contain multiple individual unrelated data transactions.

During a typical workday at an admin unit, several unit diary clerks input data changes into their UDMIPS application. Each clerk constructs a single unit diary file with multiple transactions. At some point, the diary is closed (no more transactions). Then the admin chief (senior enlisted) and admin officer review the diary entries by comparing source documents (the data request with proof of valid data or signature of requesting official) with the diary entries and proofread them for correctness. Once reviewed, the admin officer "certifies" the diary and "locks" it. Locking the diary makes it un-editable by anyone but the certifying officer. Only the certifying officer can "unlock" it for further editing if necessary. By mid-afternoon there are probably several certified diaries ready for submission. The certified diaries are transformed into their final MCTFS friendly format through the "Make Courier" process and then transmitted to a collection server.

The diaries transmitted by the admin unit are typically routed through an intermediate server at a regional administrative unit called the Marine Information Systems Support Office (MISSO). There are about a half dozen MISSO offices in the Marine Corps, typically one at each major Marine Corps base. The unit diaries from all of the administrative units are collected on a server at the MISSO, where they are checked electronically for correctness. Those diaries with bad data or improper format are returned to the admin unit for reprocessing. Diaries certified as valid at the MISSO are then transmitted in a batch to a collection server attached to the MCTFS mainframe in St Louis, MO.

In the middle of the night (five times per week: Sunday – Thursday), all diaries waiting on the MCTFS collection server are uploaded into the MCTFS mainframe and are processed. The intricate MCTFS COBOL programs process the data in the unit diaries and make data updates to the Central Master File for valid data transactions. Any unit diary transaction that fails to have the correct format, data fields, codes, or violates some encoded policy are not processed. These refused transactions are collected and returned electronically to the original admin unit that submitted the unit diary in the form of a "Unit Diary Feedback" (DFR) report. Admin units review this report for the previous day's unit diaries to make any corrections and resubmit the unit diary.

The processing of all submitted daily unit diaries on the MCTFS main-frame is called "The Cycle." After the cycle is run, the CUDDB (the local replicated version of the Central Master File at the base or station) needs to be "refreshed". In other words, all of the data updates (unit diaries) that were processed during the cycle are now present as updated data in the CMF. The CUDDBs at the local level are now out of "sync" with the CMF. The MISSOs at the local base run an automated program to extract the updated manpower data from the CMF for the units aboard their base. The extracted updated data is placed into a file called the "transaction reconciliation file" or TRECON. The TRECON is then, in turn, used to update the CUDDBs that each individual admin unit uses for daily data access. This completes the unit diary data cycle in that the data updates (unit diaries) created by the unit diary clerks using UDMIPS application is now reflected in the single source of Marine Corps manpower data (MCTFS) and is reflected in the CUDDB, the database used at the local level to access MCTFS data through the work day. Now the unit diary clerks will see the data updates reflected in UDMIPS when they pull in data from the CUDDB during the next business day. It should be noted, however, the TRECON process is notoriously unreliable, and often manpower data viewed in UDMIPS might be days or even weeks before data updates are reflected in the CUDDB.

The figure below depicts the logical workflow of the unit diary process as viewed from an administrator's perspective. Figure 5 depicts the unit diary process from a computer system perspective.

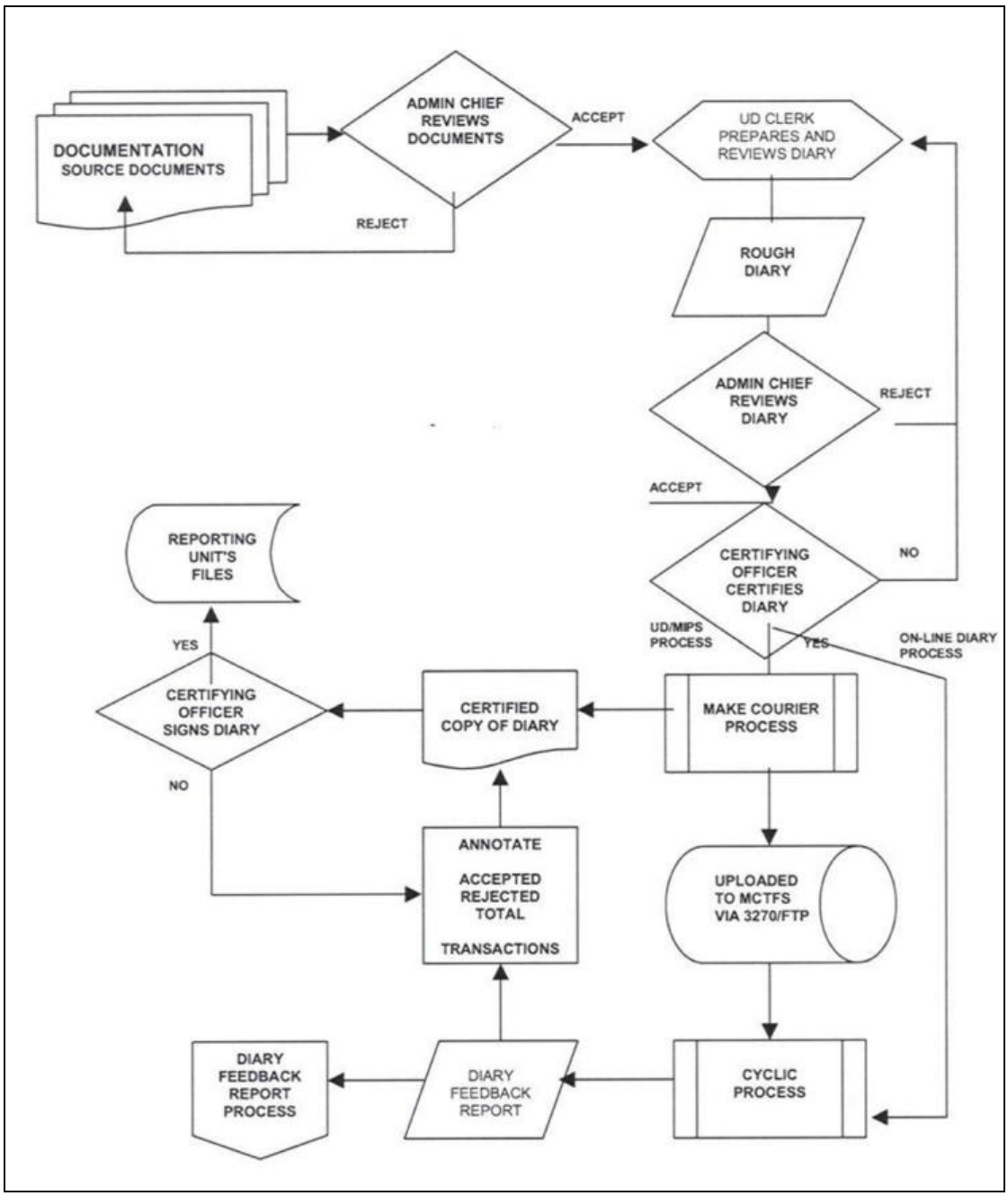


Figure 4. Unit Diary "Data Cycle" – Administrator's View [From: Ref. 1]

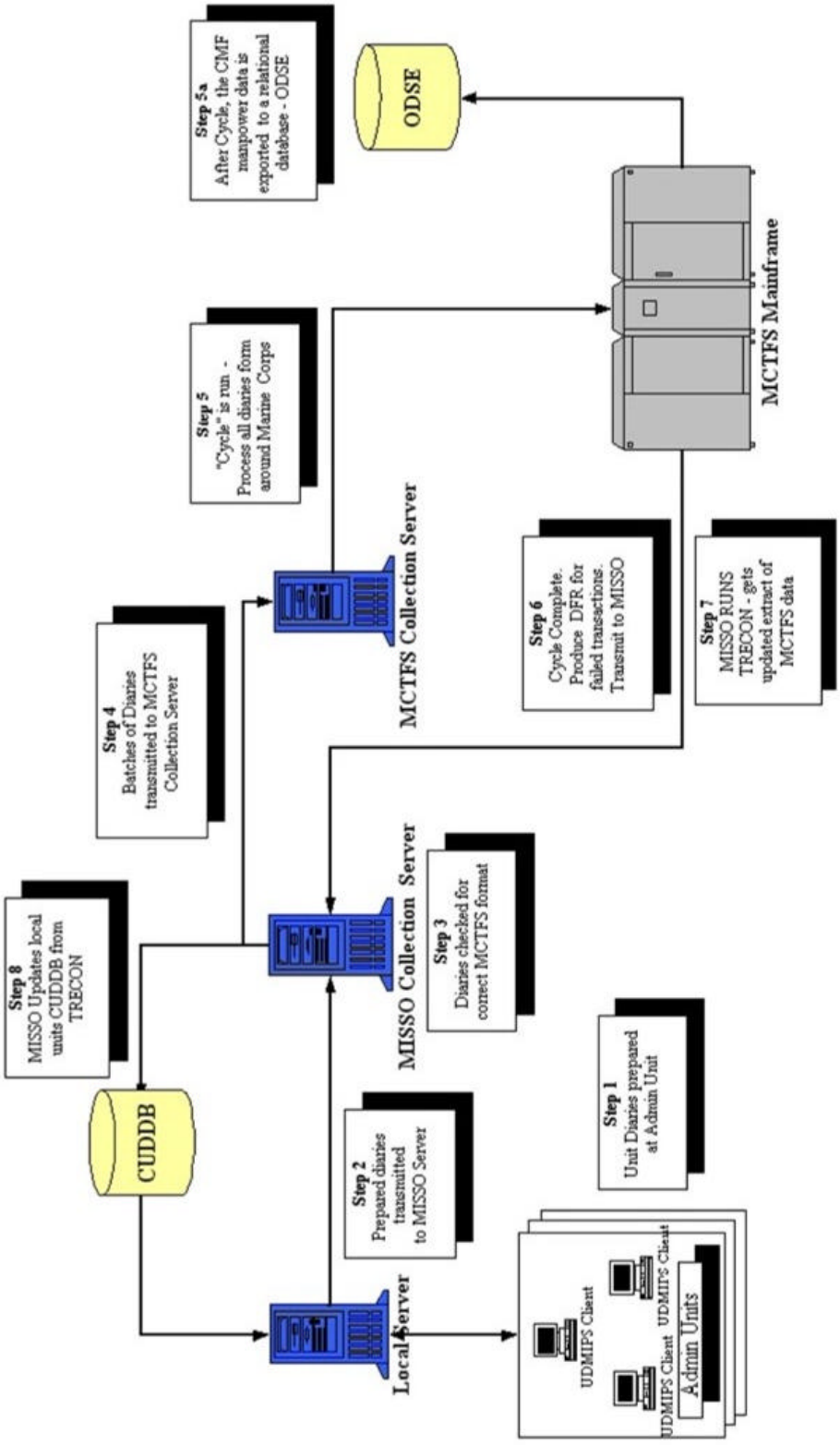


Figure 5. Unit Diary "Data Cycle" – Systems View

D. OPERATIONAL DATA STORE ENTERPRISE

As previously described, the single source of manpower data is kept in a proprietary flat file system on the MCTFS mainframe. The local slices of the MCTFS manpower data, the CUDDBs are also in this format. Thus, both the MCTFS CMF data and the CUDDBs are unreadable except by the proprietary MCTFS programs. The MCTFS system was designed before the widespread use and availability of commercial relational databases. Because this data is in a rather cryptic flat file format, it is impossible to perform the kinds of data manipulations on the MCTFS data we can now easily perform on a relational database. Over the past decade, the Marine Corps realized the shortcomings of the MCTFS data format and decided to develop the Marine Corps Manpower Operational Data Store (MCMODS). MCMODS consists of two entities the Total Force Data Warehouse (TFDW) for historical data and the Operational Data Store Enterprise (ODSE) for current data. [Ref. 4] The ODSE is a commercial relational database and is presently an Oracle 8i Enterprise database.

After each nightly cycle of unit diaries being processed on the MCTFS mainframe, a separate MCTFS program is then processed. The purpose of this program is to "refresh" the ODSE with the data updates that occurred during the MCTFS cycle. Only "touched" or updated records in the CMF are written out to the ODSE each night. [Ref. 4] This process is depicted as Step 5a in Figure 4 above. It should also be noted that the entire ODSE is refreshed whenever there are new software releases (programmatic updates) to the MCTFS mainframe, which occurs twice yearly.

The ODSE is maintained in St Louis, MO along with the MCTFS mainframe, but several replicated copies of the ODSE database are maintained at most major Marine Corps installations by the local MISSOs. The local copies of the ODSE are used as a read-only relational database data source for a wide variety of queries and reports. The desktop application available within most admin units and higher command level agencies is the COGNOS Impromptu application. Impromptu is a Windows based interactive database query and reporting tool that allows users the ability to quickly and easily create reports. [Ref. 4]

The significance of the local ODSE database is that they can be accessed via an Open Database Connectivity (ODBC) network connection. Open Database Connectivity is a standard database access method developed by Microsoft Corporation. The goal of ODBC is to make it possible to access any data from any application, regardless of which database management system (DBMS) is handling the data. ODBC manages this by inserting a middle layer, called a database driver, between an application and the DBMS. The purpose of this layer is to translate the application's data queries into commands that the DBMS understands. For this to work, both the application and the DBMS must be ODBC-compliant -- that is, the application must be capable of issuing ODBC commands and the DBMS must be capable of responding to them.

This type of data access to the MCTFS manpower data via the ODSE will allow a programmer to write web scripts (Java Servlets, ASPs, PHP, etc.) that can read out data from the ODSE and display it via the Web. The ODSE will be the key to developing the TFAS, the goal of which is to web-enable MCTFS data transactions. In the next chapter, we will discuss the general concepts of the TFAS program and it’s integration with MCTFS.

III. TOTAL FORCE ADMINISTRATION SYSTEM

From the background information about the current Marine Corps manpower data system (MCTFS and related proprietary applications), it is clear that the Marine Corps needs to reengineer its personnel administration policies and systems in order to better serve the needs of individual Marines and leaders. Over the past decade, in light of the exponential growth of the Internet and sophisticated commercial-off-the-shelf (COTS) software applications, as well as the decreasing cost of desktop computer hardware, the Marine Corps came to the conclusion that such a reengineering of administrative processes and systems was possible. The effort to leverage current and future information technology (IT) to increase data manpower data access, lower administrative personnel manning, increase efficiencies, and decrease the manpower intensive administrative processes has evolved into a funded procurement program called the Total Force Administration System or TFAS. The purpose this chapter is to outline the basic concepts, structure, requirements, and current state of the TFAS program.

A. GENERAL DESCRIPTION

The conceptualization of TFAS began in earnest in 1997 and 1998 when the general officers of the Marine Corps decided to reduce the overall manning of Marine Corps administrators by 1000 personnel as part as a service-wide manpower restructuring. [Ref. 5] Their reasoning was that technology had evolved to a point that gains could be made by automating many of the manpower intensive administrative processes. It took several more years to get TFAS established as a funded program at MARCORSYSCOM. In Fiscal Year 2000, TFAS entered Phase 0 – Concept Exploration and a series of organizing procurement documents were published. Ironically, this was the same year that the manpower cuts in administrative units took place and battalion-level admin units were consolidated to serve multiple units. To understand the TFAS concept, it is beneficial to analyze the information available through the initial procurement documents.

The description of the TFAS program as given in the Statement of Work (SOW) is as follows:

> TFAS is the technical implementation of the Marine Corps' upgrade of human resource system payroll and personnel administration services concept. This concept and technical architecture seeks to reorganize current business processes; organization structure; implement new policy and procedures; and to align information technology (IT) assets around a data-centric environment. The ability to communicate, share, and manipulate large amounts of data across a distributed operational environment is the key tenet behind this concept. [Ref. 6]

This is a very broad mission statement for the TFAS. In more specific terms, the idea behind TFAS is to use current and emerging IT technologies to improve the way the Marine Corps accesses MCTFS data. The primary focus is to web-enable MCTFS data transactions traditionally carried out through the current single port-of-entry, the administrative unit through the OLDS and/or UDMIPS MCTFS application sub-systems. In addition to developing web sites to serve individual Marines and all levels of command, the TFAS charter directs that the following technologies be evaluated and possibly implemented in TFAS [Ref. 7]:

- Interactive Voice Response (IVR)
- Computer Telephony
- Public Key Infrastructure
- Smart Card
- Personal Digital Assistant (PDA)
- Wireless Connectivity

B. THE FIVE DIMENSIONS OF TFAS

A more descriptive outline of TFAS and its impact on our manpower and pay processes is given by LtCol Jeffery Peterson, a former TFAS Branch Head, Manpower Plans and Policies, Manpower and Reserve Affairs (M&RA). In a paper published in 1999, he outlines the five dimensions of TFAS, and how the Marine Corps will change if TFAS is fully implemented. [Ref. 5]

> Role of the Commander. Battalion and squadron commanders will continue to have the capability to provide the full range of pay and administrative support to the individual Marine. Decentralized reporting and accessing of information by the individual Marines and small unit leaders will, however, change the focus of

the commander and his staff from volume transaction reporting to situational awareness, decision-making and the handling of the more technical pay and administrative processes.

Role of the Marine. Individual Marines will no longer be passive bystanders who must wait on others to conduct administrative business. Through telecommunications services, Marines will be empowered to view information and submit transactions that will generate necessary feedback reports to the commander.

Organization. The Marine Corps will eventually migrate to not fewer than three personnel administration centers (PACs), one of which will house a self-service call center. The PACs and Call Center will operate 24 hours a day, 7 days a week. Commanders at the battalion and squadron level will retain a small cell of Marines who can collect, perform quality control and submit transactions to the PACs for review and certification. Decentralized reporting by individual Marines and small unit leaders is expected to markedly reduce the number of transactions that must flow from the unit commander through the PACs.

Processes. Processes will be configured to give Marines and small unit leaders maximum visibility and access to their pay and personnel information, balanced, of course, against the need to control fraud, waste and abuse. 'Point of action/point of reporting' and single data entry processes will replace redundant handling and reporting of information. The intent is to eliminate unnecessary intermediaries who do not add value to the information being reported or accessed.

Technologies. Marines will use menu-driven, web-based technologies along with an interactive voice response system to input and access information. Smart Card technology will facilitate user identification and signature requirements. Portable electronic devices will allow for the remote capture and reporting of information and allow for information access in expeditionary environments. State-of-the-art security protocols will help prevent electronic/asymmetric attacks. Plain language Text will replace computer code and technically oriented help screens.

C. TFAS BASELINE REQUIREMENTS

A delineation of the function and form of TFAS is given in the TFAS Baseline requirements developed by the Information Technology and Infrastructure Section of MARCORSYSCOM (Program Manager for TFAS). This list of requirements will be used as a "baseline" for developing the components of TFAS. Any prototype developed for TFAS must meet these requirements. Since the focus of this thesis is to develop a

working prototype web site for TFAS, Echelon II, these requirements must be considered. The following three tables outline the baseline requirements.

FUNCTIONAL BASELINE REQUIREMENTS
Reduce the overall cost of administrative functions and reduce administrative structure for reallocation Marine Corps-wide.
Serve as comprehensive Human Resources Management System, not limited to pay and routine personnel administration.
Increase the Marine Corps' operational efficiency and effectiveness through improved Quality of Life (QOL) and decreased administrative footprint.
Streamline current administrative processes with an eye toward automation and compatibility with evolving DoN/DoD related initiatives.
Provide pay and administrative services to Marines and commanders who operate in an expeditionary, "reach back" environment.
Provide service capability in all Marine Corps operating environments. Operate independent of location. Service will be available from home, in garrison, while traveling, embarked aboard ship, or deployed.
Provide self-service capability to the individual Marine.
Provide 24/7 service routine processing capability.
Develop processes and systems that are scalable; i.e., not dependent on the number of Marines being supported or the number of Marines providing the support.
Processing data. Small unit leader and training managers must be able to collect, pass, and report pay and personnel information from the point of action.
Download access. Small unit leader and training managers must have electronic download access to pay and personnel information on their Marines.
Eliminate the Service Record Book (SRB). All pay and personnel information must be stored electronically in the Marine's pay/personnel record in MCTFS.
Minimize training requirements for operators. Technical knowledge, rules and edits must be built into the system to minimize the training requirements for operators. This also includes maximum use of plain language vice codes.
Reduce redundant data entry. Administrative databases electronically linked.
Enable event driven automated processes.
Shift responsibility for data entry for personnel transactions from the administrator to the individual.
Improve customer/client satisfaction through ready access to information.
Reduce the cost of transactions.
Improve the quality of services rather than concentrate on the individuals who provide the service.
Provide readily accessible personnel and payroll data to Marines and commanders.
Implement business applications to reduce the labor-intensive processes and realize a labor cost savings.
Provide an improvement in automated data processing in the following areas: data entry redundancy, data entry errors, processing speed, and data analysis.
Provide self-service technology to give more control to the individual for routine transactions.
Individual Marines, unit administrative offices, consolidated training facilities, unit training offices, and other higher headquarters agencies must be able to report transactions through a distributed architecture.
Provide information security safeguards to resist and otherwise prevent electronic attacks from known and suspected asymmetric/electronic threats, and the prevention of fraud and other system misuses and abuses.

Table 2. TFAS Functional Baseline Requirements [From: Ref. 8]

PERFORMANCE BASELINE REQUIREMENTS
Serve the Total Force (Active, Reserve, Retiree, Family Member) through a web-enabled system that can provide Marines the ability to conduct secure administrative processes wherever they have access.
Embrace Smart Technologies to facilitate the collection and reporting of pay and personnel information by allowing Commanders to collect and enter information at the point of action and then upload it to the Marines Corps' attendant systems.
Enable Information Technologies and select those that serve not only today, but also future technologies (e.g., wireless).
Individual Marine capability. Marines will have the capability to submit pay and personnel account transactions via a web based, menu-driven application from a computer with Internet access. This capability will be available from ships and the full range of expeditionary environments. Authentication and verification procedures are required. Confirmations of the transaction will be sent via the web application, provided via the telephone or will be mailed to the Marine.
Capability of the commander. Commanders at the battalion/squadron and above level will retain an administrative capability to collect, provide quality control and forward personnel and pay related information to the Service Centers for final processing. This will be primarily focused on command-originated data, but the capability to forward any information on behalf of the individual Marine will be available. These administrators will also review feedback reports on data submitted. The commander will have access to pay and personnel related information.
TFAS will encompass "centers of expertise" which will provide non-routine and specialized full service capability 24 hours a day, seven days per week.
Single Data Entry. Data inputs into data templates or form based formats must automatically generate the updates to the master electronic record.
Automate, via the Internet, pay and personnel administrative support.
Enhance information security capabilities to prevent system misuse using "state-of-the-art", non-developmental identification and authentication technologies.
The TFAS technical architecture environment will be: secure, responsive, reliable, flexible, maintainable, interoperable, scalable, easy to use, affordable, and survivable.
Provide unit commanders on-call tailored reports and analytical data.
Enable batch processing for personnel administration transactions.
Permit data entry from its sources and reduce manual transaction routing. Examples include PFT and marksmanship scores entry from testing site.
Remote access. Marines must be able to review pay and administrative information and conduct associated transactions without having to be physically collocated with the service provide. The remote access (input/output) capability must be available at the lowest level possible.
Authentication. The system must authenticate user's identities.
Digital Signature. The system must allow operators to "sign" documents in a paperless environment.
Verify transaction. Marines must receive verification of their completed transaction with an estimated date on which that transaction will affect the Marine's pay or other record.
Self-service access (input/output) must provide for routine transactions by the individual Marine on personnel and pay related matters.
Pending actions. Marines will receive advisories where supporting documentation is required to complete an action.
Command notification. Commanders must be informed of those transactions that are submitted through a self-service/Call Center capability.
Tailored "on call" feedback reports for unit commanders will be available on all transactions of Marines within their authority as well as alert them to various information or action required relative to their Marines.
Commanders must be able to collect, QC, and forward pay and personnel information to center for

PERFORMANCE BASELINE REQUIREMENTS
expertise (e.g., call center) for final review, certification and processing.
Direct data input. Pay and personnel information forwarded from commanders must not require re-keying or manipulation for data base entry.
Open architecture. Consolidated service centers must be able to receive and process transactions from the commanders they support as well as any commander in the Corps in case of contingencies.
Unit commanders must have redundant communication links through which they can forward preformatted pay and personnel information to Consolidated Service Center(s).
Enable multiple transaction processes triggered by one administrative request.
Shared processing centers to complete "back office" activities resulting from self-service or call center transaction.
Center of expertise to provide non-routine and specialized assistance.
Transactions and authentication traceable to the source.
Utilize existing Commercial-Off-The–Shelf (COTS) and Non-developmental Items (NDI) as much as practical.
Port of entry to MCTFS must be at the lowest level possible and not require intermediate processing.
Contribute to an "operator friendly" environment in that the operator is not expected to be a technical expert in any matter. Provide for ease of manipulation of data within the system with "Help Screens" available while navigating the system.
Provide data source tracking.
System must maintain an historical record of all automated transaction processing.
System must provide immediate confirmation of all transactions submitted for processing.
Establish "Call Center(s)" to support commanders and the individual Marine regardless of time and location.
Transactions requiring certification/authentication from higher authority or those requiring supporting documentation will be held in a "Pending File" status. Those files will have an advisory forwarded to the applicable Marine on a regular basis before the transaction is carried forward for processing or discarded to a "Pending Status" expiration.

Table 3. TFAS Performance Baseline Requirements [From: Ref. 8]

INTERFACE BASELINE REQUIREMENTS
TFAS will interact with current and future Marine Corps and Joint data systems, tools, and repositories, including the Defense Integrated Military Human Resource System (DIMHRS) and the Marine Corps Total Force System (MCTFS).
TFAS computer/information technology will be synchronized with related Marine Corps infrastructure and technology initiatives and upgrades.
TFAS will establish a logical migration path that supports DIMHRS rather than being a "new start" that may have conflicting and competing requirements with DIMHRS.
Remote data capture/upload/download access capability. Commanders and unit training managers will have the capability to capture, store and upload information at the point of action. The operator should be able to populate the data fields through more than one means (e.g., download from UP/MIPS or its equivalent, download from the Operational Data Store Enterprise (ODSE) or MCTFS, download from a smart card, or manual entry). The same type of redundancy must exist for uploading information.
System must be capable of providing connectivity through the Marine Corps tactical data network, Unit Operations Center (UOC), in a deployed environment.
System must be scalable in the acceptance of authentication technologies in development, e.g., Smart Card, PKI, and electronic signature.

Table 4. TFAS Interface Baseline Requirements [From: Ref. 8]

D. TFAS ECHELONS

The implementation of TFAS is not as simple as mapping the above baseline requirements to technical requirements, writing some code and setting up a web site. The technical implementation of TFAS is only half of the equation. Full implementation of TFAS will require the reengineering of processes and writing of new policies to support TFAS. For this reason part of the founding architecture of TFAS encompasses how Marines at different levels of command will interact with TFAS. This architecture is organized into five echelons of users. They are:

Echelon	Entity Supported	Description
I	Individual Marine	-Perform routine, self-service transactions -Review pay & administrative information -Responsible for keeping commander aware of pay and personnel information changes
II	Small Unit Leader	-Focus is at Company Level -Limited UD/MIPS reporting capability -Retrieve pay & personnel information -Produce rosters & reports -Enhanced situational awareness & decision making capabilities
III	Battalion/Squadron	-Personnel Administration Reporting -Responsible for accurate & timely reporting of pay and administrative information -Maintain ability to report unit diary entries and generate reports -Serve as interface with Personnel Administrative Center -Improve situational awareness and decision making ability -Retain a cell of administrative personnel
IV	Personnel Administration Centers	-PACs serving multiple Battalions -Establish no fewer than 3 Regional Personnel Admin Centers -Establish one Call Center providing technical support 24/7 -Report & certify pay & administrative information not reported elsewhere
V	HQMC	-Retrieve pay & administrative information on subordinate organizations -Maintain responsibility for CRUC & AOWP -Submit queries & retrieve information from ODSE -Retain stand-alone reporting capability

Table 5. TFAS Echelons [After: Ref. 9]

Currently, all administrative functions funnel through one point-of-entry: the administrative unit (PACs) and, ultimately, the overworked unit diary clerk. The goal of the echelons of TFAS is to partition the work to the appropriate level. Figure 6 below depicts the operational architecture of how the work will be partitioned by echelon.

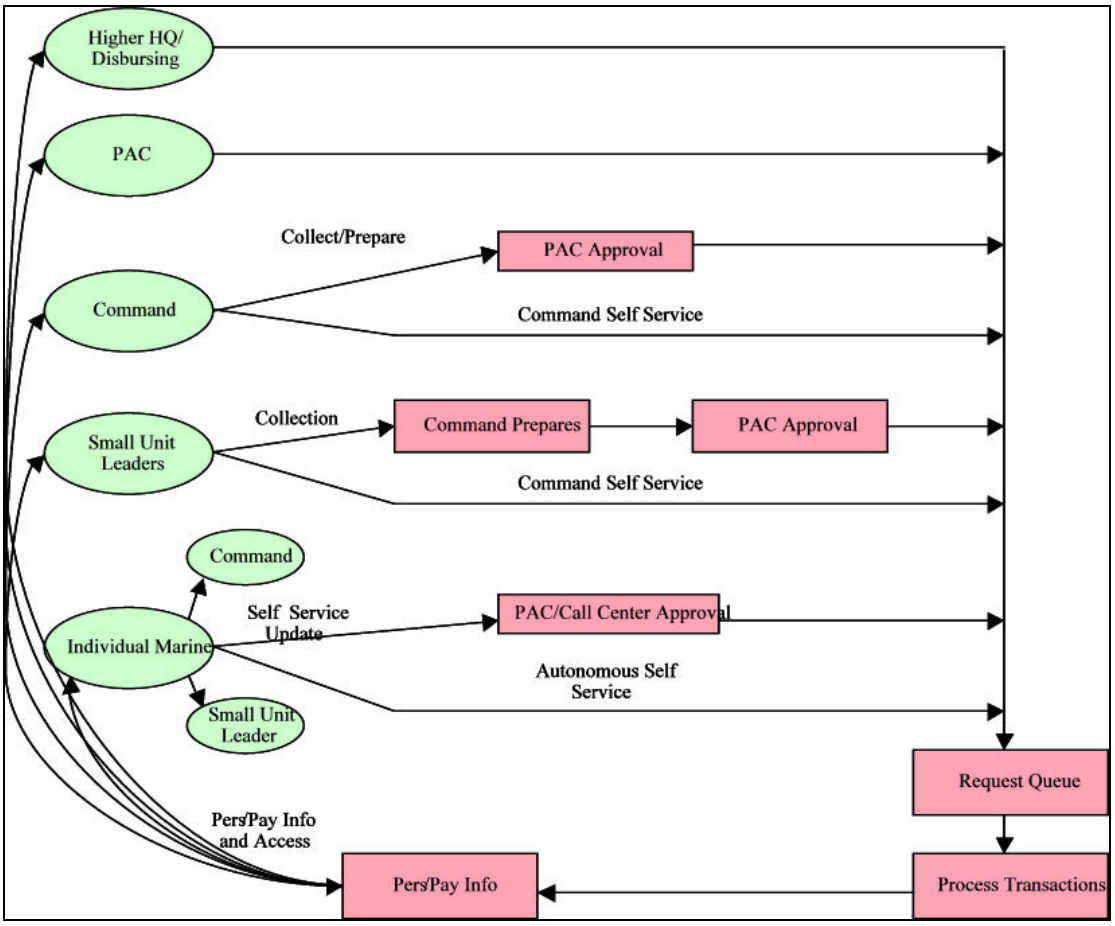


Figure 6. TFAS Echelons - Operational Architecture [From: Ref. 9]

E. TFAS, ECHELON II

TFAS as a procurement program is in the concept exploration stage of development. As a result, the detailed technical architecture of TFAS is presently undefined. There are, however, conceptual models for the technical architecture for each echelon. As this thesis is focused on Echelon II, a study of this model is appropriate. The models of the other echelons are not presented, although they are similar in nature yet tailored to the requirements of the particular echelon. Figure 7 below depicts the conceptual technical architecture for Echelon II.

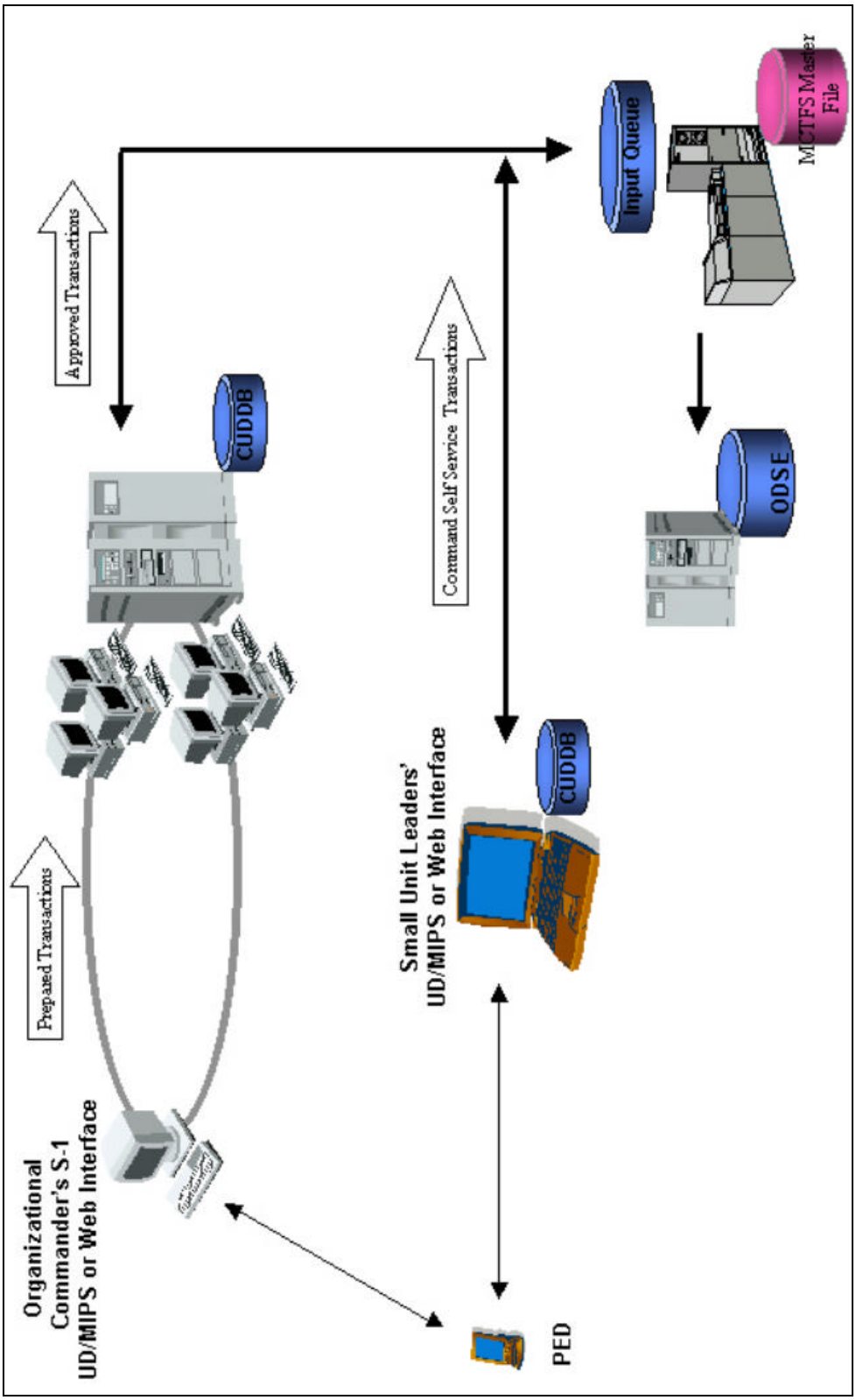


Figure 7. TFAS, Echelon II – Conceptual Technical Architecture [From: Ref. 9]

Figure 7 shows that the small unit leader may interact with TFAS and thus MCTFS data in several ways:

1) Directly via a desktop workstation utilizing a web interface or UDMIPS.

2) Directly via a personal digital assistant (PDA) that is linked wirelessly to the internet.

3) Indirectly through the leader's chain of command by requesting administrative transactions at the next TFAS echelon (Echelon III Battalion/Squadron).

This diagram is misleading, however, because it depicts the small unit leader possibly interacting with UDMIPS. From the information presented in Chapters 1 and 2, however, we believe that this idea would violate many of the baseline requirements like ease-of-use, require little training, intuitive interface, etc. Figure 7 is accurate in its' depiction of the conceptual flow of data from end-user to the single source of data, the MCTFS CMF. It is this conceptual architecture upon which we will base the Echelon II web site prototype design.

F. CURRENT STATE OF TFAS DEVELOPMENT

The information presented in this chapter represents a concise summary of the state of development of the TFAS program. As a program that is still in the concept exploration phase, only concepts and broad functions have been defined. Efforts underway by the three interested agencies: 1) MARCORSYSCOM, 2) M&RA, and 3) DFAS (TSO) have focused on the development of Echelon I (Individual Marine). Detailed system and user functional requirements have been defined and published for Echelon I. Additionally, the development of Echelon I has proceeded abnormally briskly because of an existing web site named Marine OnLine (www.mol.usmc.mil) or MOL. MOL has been around for several years and is now being used as a baseline for TFAS web site development. Currently, an individual Marine can setup an account and access various read-only reports pertaining to the Marine's personal MCTFS manpower data. TFAS developers have adopted this existing web site and are currently working on implementing the "write" functions defined in the Echelon I user functional requirements. If successful, the MOL web site will enable the individual Marine to actually conduct

some limited MCTFS data transactions traditionally performed through the administrative unit. The following figure depicts the user interface for MOL for a user viewing his/her personnel training data.

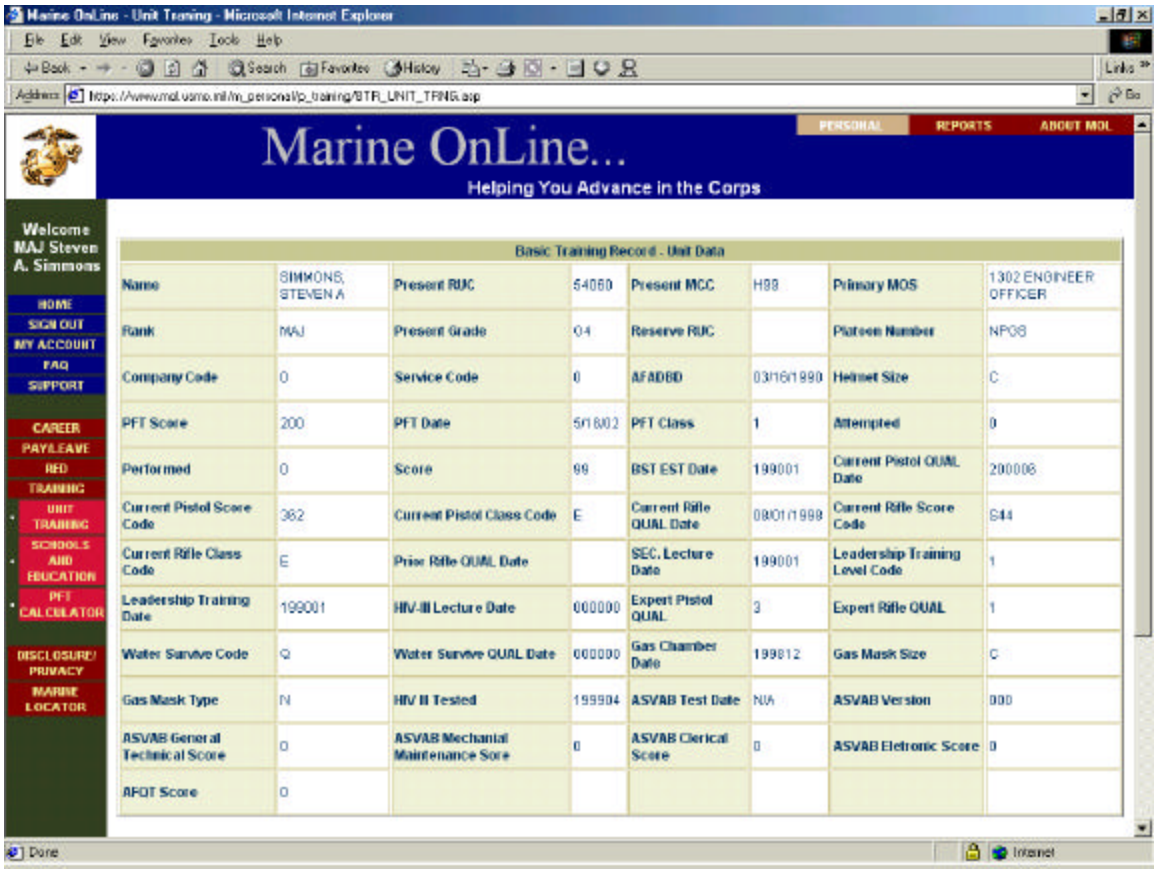


Figure 8. Marine OnLine User Interface

The data displayed to the user is pulled from a replicated copy of the ODSE mentioned in Chapter 2. The web scripting used by MOL is the Microsoft-based technology, Active Server Pages, running on a Microsoft web server, Internet Information Server-5. TFAS developers have stated that MOL will eventually serve as the universal portal for all echelons of TFAS. This TFAS, Echelon I prototype demonstrates that the Marine Corps has "unofficially" adopted the Microsoft model of technology for the development of TFAS. It is for this reason and the tenets of the NMCI (Navy-Marine Corps Intranet), which also has adopted Microsoft based technologies, that we have developed the Echelon II web site prototype with Microsoft products.

As of April 2002, TFAS developers at TSO had a working prototype that allowed an individual Marine to conduct a few data transactions via MOL. Testing and refinement is still currently underway and, thus, this capability is not yet deployed publicly. Their efforts, however, pointed us in the proper direction for the development of the prototype with respect to "write" transactions. The transactions are made possible by taking user input via the web and creating an XML document that emulates a unit diary that can then be processed on the MCTFS main-frame like normal unit diaries.

Beyond the efforts to get Echelon I up and running, all other echelons of TFAS remain undeveloped. There are no detailed user or system functional requirements defined for Echelon II that can be used to develop a detailed system architecture or code for web pages. The remainder of this thesis will present our interpretation of the broad concepts defined for the TFAS, Echelon II web site and how it will interact with MCTFS. The goal of this analysis is to develop a working prototype and define in specific terms "a solution" for the next Echelon of TFAS. This analysis begins in the next chapter by defining the functional requirements of the prototype.

IV. FUNCTIONAL REQUIREMENTS OF TFAS, ECHELON II

In the development of any software application, before any programming can be done, the requirements for the project must be as clearly defined as possible. As described in Chapter 3, the TFAS program has only been defined in broad conceptual terms and no detailed overall system architecture has been described. Additionally, no detailed functional requirements for Echelons II - V exist. Therefore, in order to develop a working prototype for the TFAS, Echelon II web site, we must define these items. The purpose of this chapter is to define, in non-technical terms, the functional requirements of the TFAS, Echelon II web site.

A. METHODOLOGY

Normally, in the course of software development, the project is defined by the needs of the customer. Typically, a project team member would act as a liaison between the programmer and the customer. This liaison would research the customer's environment and conduct extensive interviews with employees of the customer in order to develop the functional requirements of the project. These requirements would serve as a bridge between the customer and the programmer. The requirements are written in non-technical terms so that the customer can understand them, and, eventually agree they adequately define the desired functions of the software application. The requirements should also contain enough detail and structure so that the programmer can use them to begin coding. Once various software modules are developed, the customer then reviews the working prototype. Upon this review, the customer will often submit changes to the requirements because the product does not meet their expectations, there are ill-defined requirements initially, or new additional requirements are identified. This process of module development, review and requirements redefinition is an iterative process that can cycle many times until the final product is deployed. The prototype web site developed for this thesis will only undergo one cycle of this design process.

In developing the prototype for the TFAS, Echelon II web site, the "actors" in the software development process do not exist. We will serve as customer, liaison and programmer. For this reason, the requirements defined for the TFAS, Echelon II web site

will be based largely upon my own experience as Marine Officer and my three tours as a company commander. Since Echelon II is focused on serving the needs of the small unit leader, which by definition is centered at the company command level, a good estimation of the "customers' needs can be defined. To aid in this development, the existing manpower data transaction functions resident in UDMIPS that normally serve the data requirements of the company will also be considered. Most certainly, the requirements defined in this thesis will not be the final requirements for TFAS, Echelon II. The final requirements will have to be vetted through several levels of decision makers at M&RA and MARCORSYSCOM and ultimately approved by the Milestone Decision Authority (MDA) for the TFAS program. It is expected that many of these decision makers will want to add additional functionality or remove a certain data transaction because they feel the company commander should not have the authority to conduct that transaction. While this is expected, the decision makers should be warned to review the functional requirements of TFAS, which states that TFAS should "shift responsibility for data entry for personnel transactions from the administrator to the individual."[Ref. 8] This functional requirement for TFAS is an extremely important guiding principle in that we must appropriately partition the administrative data transactions presently residing solely in the PAC to the appropriate TFAS echelon. If we fail to fully empower the users at each echelon because we, as an organization, are fearful of change, we will never remove the administrative chokepoint in the PAC, nor realize the efficiency gains that are envisioned for TFAS.

The functionality of the TFAS, Echelon II prototype developed for this thesis will only encompass "atomic" transactions. Atomic transactions are those data transactions which only involve a single step – that of one TFAS user inputting data via the web and, once submitted, the data being transformed into a unit diary which is processed on the MCTFS main-frame. Non-atomic transactions are defined as those data transactions where multiple TFAS users within one or more echelons must be involved in the transaction over a period of time before a unit diary is created and transmitted for MCTFS processing. The development of these non-atomic transactions in a working prototype would require that multiple echelons be present. This functionality is beyond the scope of this thesis. However, the final deployed TFAS, Echelon II web site will

have to encompass non-atomic transaction functionality. The functions defined in this thesis will only pertain to atomic transactions at the Echelon II level.

It is from this perspective that we will define the functional requirements for the TFAS, Echelon II web site. The functional requirements will be broken down in the following categories: Users and Roles, System Requirements, Reports, and Data Entry.

B. USERS AND ROLES

Before a definition of the functional requirements can be defined, the users at the Echelon II level must be described. In all of the conceptual documents pertaining to the TFAS program, Echelon II has been defined as the domain of the small unit leader. No guidance has been given with regards to meaning of the term small unit leader. However, this term is common in the Marine Corps and is generally accepted to mean those leaders at the company level and below. The lowest level of legal command is at the company level. The term legal, as used here, means that there is some formal responsibility associated with the command billet. For example, the company commander is appointed in writing by the battalion commander, has non-judicial punishment authority, etc. The other small unit leaders found at the company level such as the squad leader, platoon commander, and company staff members (company executive officer, 1st Sergeant, etc.), do not have this type of legal responsibility. They do however have a responsibility for the leadership of Marines at the company level and have a role to play in ensuring accurate and timely data regarding their Marines duty status is maintained in MCTFS.

Depicted in Figure 9 below is an organizational diagram of the generic command structure found in the Marine Corps along with a mapping of TFAS Echelons I – IV. Units are depicted using the organizational structure typically found in the major ground units, the Marine Division and FSSG. Units in the Marine Wing, base and support units, detachments, and other unique Marine organizations have a different structure than that depicted in Figure 9. However, these units do have a hierarchical structure similar to ground organizations and, thus, at each command level an equivalent unit usually can be identified. For example, in the Marine Wing, a squadron is equivalent to a battalion. For simplicity, units and their leader titles will be defined in terms of this ground command structure, which is familiar to all Marines.

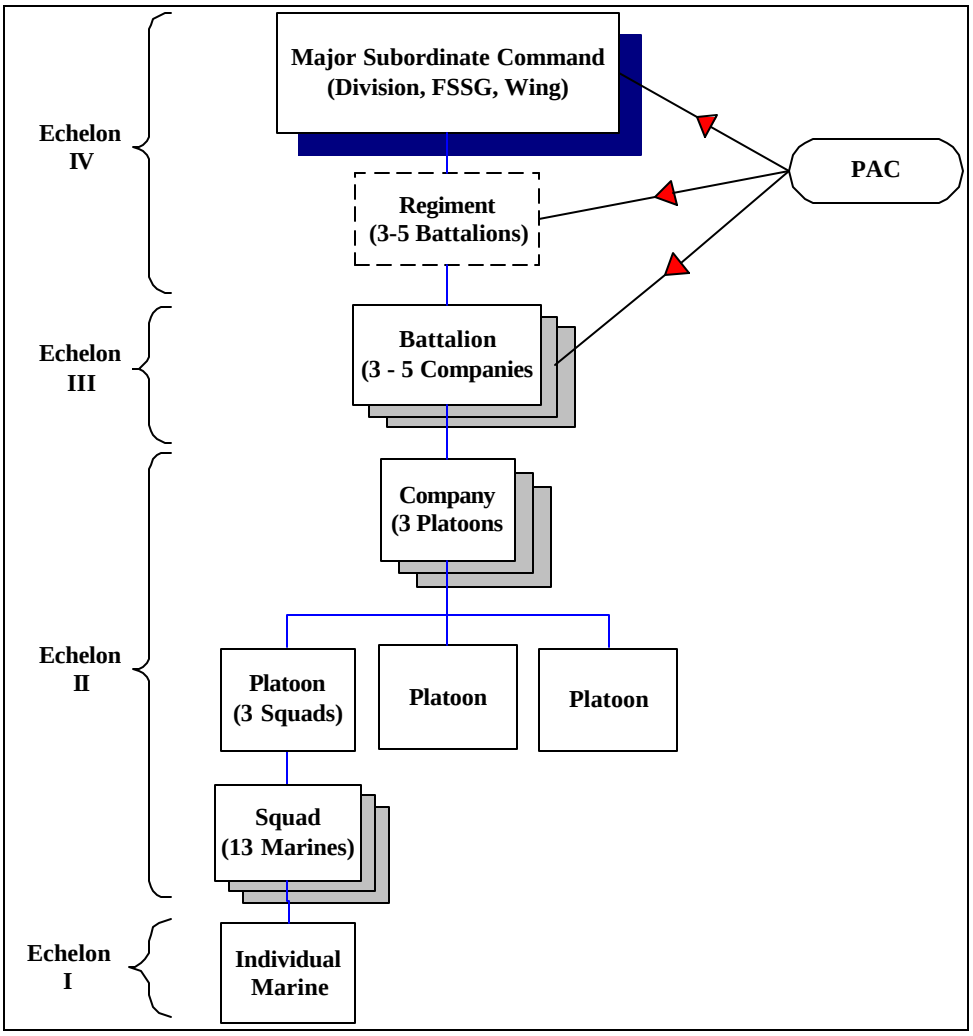


Figure 9. Command Structure and Echelon Users

1. Users

With the focus of Echelon II being at the small unit level, potential users of the TFAS system could include the following users:

- Company Commander
- Company Executive Officer
- Company 1st Sergeant (Senior Enlisted)
  - Company Clerk(s)
- Company Gunnery Sergeant
- Company Training NCO/Clerk
- Platoon Commander
- Platoon Sergeant
- Platoon Guide
- Squad Leaders

2. Data Flow and Authority

Presently, all of these small unit leaders have a role in maintaining accurate data in MCTFS. A request for data is typically generated at a very low level (individual Marine or squad leader) and is passed up the chain of command where each leader makes a decision and/or adds information. At some point, through the company commander and his/her staff, the data is approved as valid and is sent to battalion for further approval and handling. Battalion then passes the data to the PAC where it can be further scrutinized and approved as valid. Finally, the data is given to the unit diary clerk to be entered into UDMIPS and a unit diary is created for the data transaction.

With any implementation of TFAS, this business process must be changed. For Echelon II transactions, the data flow will proceed as described above, but will stop at the appropriate small unit leader or designated company clerk where the data will be entered via the web. Additionally, requests for data (reports) no longer have to be processed through multiple individuals. The vision is that the small unit leader will have direct access to the data he or she is authorized to view. For example, a platoon commander has every right to directly access the data on the Marine under his/her charge without requesting this data through several layers of bureaucracy.

The sticky question for the decision makers on the final form and function of TFAS will be what specific actions are authorized to a particular leader. For example, who should be able to directly enter a PFT score into the TFAS web site...a platoon commander…company commander...or battalion staff? As these decisions are formulated, the TFAS programmer will have to then implement these policies in programming code. For instance, maybe it will be decided that only the members of the company staff can enter PFT scores. This would mean the web page for performing this function would only be accessible to those users. Platoon and squad users would be denied access. Wrangling over these decisions will be a source of delay in the deployment of TFAS and its intended benefits.

3. Roles

In our view, there are several types of users at the small unit level. They are defined by the level of authority to read or write data on the TFAS, Echelon II web site.

Role #	Role Name	Description
1	Full Authority	Read and Write permissions on all functions.
2	All Data Read-Only	Read-Only permissions on all reports.
3	Training Input	Read-Only permissions on all reports and limited write permissions for training data.
4	Limited Read	Read-Only Permissions on selected reports.
5	Limited Write	Write-Only Permissions on selected data transactions.
6	Limited Read and Write	Combination of Roles 4 and 5.

Table 6. Roles of Echelon II Users

Implementing these roles in the TFAS, Echelon II web site, will be a simple matter of associating these roles to read and write access permissions on the various web pages (files) on the web server and tables in the backend database. Permissions groups could be created for these roles and individual users can be added to these groups for ease of assignment and de-assignment over time. For special cases, individual user accounts can be given the appropriate access to certain web pages. This is not recommended, however, as management of the numerous, complex permissions could prove unwieldy. The drawback of this definition of roles is that it is fairly broad and does not specify the role each TFAS, Echelon II user will be assigned. It is not our intention to specifically define this mapping of user to roles in the prototype, but merely to set up the architecture to support such an implementation. Many TFAS and Manpower decision makers will ultimately decide the authority accorded to each type of user. Our recommendation to the TFAS decision makers is to keep it as simple as possible. Streamline permissions into 2 or 3 roles (Groups): Full Authority for company staff (Officers and SNCOs), All Data Read-Only for platoon and squad leaders and a special role for company clerks who have Limited Read and Write Role.

One aspect of describing the roles at the small unit level is the limitation to which restrictions on the viewing of data can be defined. This limitation is related to the idea that a certain type of user should only be able to view data on the Marines under his/her charge. For example, the company commander should be able to view only data on the Marines in his/her company and no data on Marines in other companies. This functionality of tying data viewing to the command level of the user is implementable at the company level and even the platoon level since there are data fields in the MCTFS database for company and platoon. These data fields can be used to form queries for data

based on the user's company and platoon. However there are no data fields for command levels below the platoon level, say for a squad in the platoon. Therefore, it is presently not possible to define a role for the squad leader where that squad leader would only be able to view data on his/her squad and no others. While the ability to define "platoon data only views" is possible through the construction of queries based a user's logon ID and the platoon code, the prototype developed for this thesis will not implement this. It should be noted however, that if the TFAS decision makers decide that platoon level views are desired, they could easily be implemented by the same means that the prototype implements company level views.

For the purposes of developing the prototype, all demonstrated users will have "Full Authority" and all views will incorporate all company personnel. Limiting access to various pages will be controlled through access permissions assigned to the individual pages rather than programmatic access control. It will then be a simple task to create TFAS user accounts or Groups and assign limited permissions for certain pages.

C. SYSTEM FUNCTIONAL REQUIREMENTS

Before defining the reports and data entry tasks that should be available to the TFAS, Echelon II user, general functions of the web site need to be defined. The system functions shown in Table 7 augment or further define the general baseline functional requirements listed for TFAS in Chapter 3.

Function	Description
Logon & Authentication	Web Site must authenticate each user. This will require the use of a system of user accounts and passwords.
Data View Restriction	The capability to restrict access to each page within the web site is required. For example, some users may only be granted read-only (reports) capability. The desire is to tie the user account (from logon) to view restriction seamlessly. Implement Roles defined above. Design should be flexible so as to modify user's access permissions to the individual web pages easily.
Account Administration	The requirement for user authentication and view restriction will necessitate the creation and maintenance of a system of user account information. This system must be accessible by the operating system, database server, and web server for tight integration. Where possible, there should be one account for any Marine when accessing a Marine network and the TFAS web site and data. Design various groups of types of TFAS users as described in Table 6 above.
Secure Connection	Data being displayed on the Web site is not "classified," but it is sensitive and private. Almost every view presented on the web site will list social security numbers for Marines within the unit. This data and other personal

Function	Description
	data of the Marines must be safeguarded and only accessed by authorized users. As such, transmissions between the TFAS client and the local TFAS web server must be encrypted for transmission over local intranet within the confines of a base or even over the internet.
Report Printing	Web pages must be designed such that there is a capability to print various reports. These reports must look professional and not be cluttered with web page controls (buttons, flashy graphics, etc.). These reports should as much as possible resemble the quality of reports that could be printed by a word processing application
Sorting	Incorporate sorting functionality on read-only reports for fields most commonly sorted like Last Name, Rank, DOR, and EAS.
Persistent Data Entry Documentation	Data entered should not only create and transmit the appropriate unit diary, but there should also be a "receipt" accessible by the user. This receipt may be a web response after data is submitted that can be printed or saved to the user's computer. This requirement can also be fulfilled by making the actual unit diary files created by the transaction emailed to the user's network email account where the user can store them on their on computer. Note: user feedback for successful transmission of data entry is a different function than persistent data entry documentation
Data Entry Feedback	Provide feedback to user after data entry is submitted to server. This could be a simple confirmation message or a printable report displaying all data entered which could be used as a hard copy receipt for the transaction.
Personalize User Access	Web site should "know" who is logged on and thus personalize the various features and functions of the interface.
Provide Links	Web site should have links to national USMC web sites and if possible local links to local parent units and base functions. A database table of these links could be maintained by the local webmaster and code could be generalized to load this list from the local database and thus customize the local TFAS web site.
Consistent Interface	Web pages for established TFAS, Echelon II functions should not vary from locale to locale. This is to ensure users around the Marine Corps can expect the same interface wherever they are in the Marine Corps and after a transfer. Updates to these standard pages should be made at a single national level and then software changes are published to the distributed local TFAS Web Severs. Recommendations for changes and upgrades can be made to local web masters, consolidated nationally, and then implemented.
Allow Innovation	Allow local users who are web and database literate to develop additional functions or to develop a prototype upgrade to an established function. This will serve to improve the web site over time and gain customer buy-in.
Keep It Simple	No large (file size) graphics or animations that make web page access slow. Design with the assumption that all users only have 56Kb data transfer speed. Make site navigation to the desired data function easy to find and access through and easy to use menu system.
Intuitive Data Entry Interface	Design Data Entry interface such that users can quickly navigate to desired Marine and input data. Develop easy to use record navigation or individual Marine search capbility.
Overall Web Page Design	Should adhere to MCO 5720.76 Standardization of Publicly Accessible Web Pages [Ref. 11] and its corollary, The United States Marine Corps World Wide Web Style Guide. [Ref. 12] These are lengthy documents, so a listing of their details here is omitted. Where applicable, I will site design decisions in the following chapters.
Backend database	The backend database for the web site must be a mirror copy of the data found in the ODSE in order to update it after the MCTFS cycle. As such,

Function	Description
	the existing schema (table structure and data types) should not be altered. This will make daily replication (refresh) an easy uncomplicated procedure. Additional fields could be added to tables for local use. Additonal tables can be added for local use.

Table 7. System Functional Requirements for Echelon II

D. REPORTS FUNCTIONAL REQUIREMENTS

Data display functional requirements are determined by the data needs of the users at the company level. From my experience as a company commander on three separate occasions, I have keen insight into the data needs of a company commander and his/her subordinate leaders. It is from this experience that these functional requirements are derived. Data display functional requirements means defining what read-only data or reports must be made available to the small unit leader. Table 8 lists these functions by report name, description, and data fields from MCTFS that need to be displayed. It should be noted that the data field names listed in the table have been modified slightly from the actual field names found in the MCTFS data schema in order to make them more understandable to the readers of this thesis. These functional requirements will be used to design the web pages of the TFAS, Echelon II prototype.

Report Name	Description	Data Fields
Individual Marine Data		
BIR – Contract Info	Basic Individual Record for an individual Marine. Standard UDMIPS report used by small unit leaders around the Marine Corps.	Rank, FName, MI, LName SSN, Present Grade, Plt Code, Co Code, Present RUC, Present MCC, Reserve MCC, Reserve RUC, AFADBD, Service Code, Component Code, EAS, Strength Category, EOS, ECS, Duty Limit, Duty Limit Date, Date of original Entry, Start Mandatory Drill Date, End Mandatory Drill Date, Length of Enlistment, Length Current Extension, Current Extension Number, Program Enlisted for, Total Months Extended, Active Duty MGIB Status, Time Lost Current Enlistment, 6 Years Obligated Start Date, Designated Military Pilot, Source of Entry, Officer Candidate Effective Date, Source of Initial Entry

Report Name	Description	Data Fields
BIR - Personal Info	More data pertaining to an individual Marine. Standard UDMIPS report used by small unit leaders around the Marine Corps.	Rank, FName, MI, LName SSN, Plt Code, Co Code, Present RUC, Present MCC, PMOS, Reserve MCC, Reserve RUC, AFADBD, DOR, Service Code, Component Code, DOB, HOR, Citizenship Code, Country of Origin, Ethnic Code, Civilian Education Level, Education Certificate, Blood Type, Sex, Religion, Home Phone, Work Phone, Home Street Address, Home City, Home State, Home Zip, Address Validation, Good Conduct Medal Date, Armed Forces Reserve Medal Date, Selected Marine Corps Reserve Medal Date, Duty Preference 1, Duty Preference 2, Duty Preference 3, Record Status, Last Screening Date, Marital Status, Total Number Dependents, Dependent Certification Code, BAS/COMRATS Code, Dependent GEO Location Code, Dependent Location Began Date
RED	Record of Emergency Data. Standard UDMIPS report used by small unit leaders around the Marine Corps. NOK or Next of Kin of Marine - The person designated by a service member as being the relative or person to notify in the event the service member is injured, missing in action or killed while on active duty. Notify is the person to notify if a Marine is injured, MIA, or deceased. MIA – Missing in Action.	Rank, FName, MI, LName SSN, Plt Code, Co Code, Present RUC, Present MCC, Reserve RUC, Reserve RUC, Service Code, AFADBD, Spouse First Name, Spouse Last Name, Spouse Street Address, Spouse City, Spouse State, Spouse Zip, Father Name, Father Address, Mother Name, Mother Address, NOK Notify Code, NOK Relationship, NOK Phone Notify Name, Notify Address, Notify Directions1, Notify Directions2, Notify Directions3, Notify Directions4, Notify Directions5, MIA Notify Name, MIA Notify 1st Phone, MIA Notify 2nd Phone, MIA Notify Relationship, MIA Notify Directions1,

Report Name	Description	Data Fields
		MIA Notify Directions2, MIA Notify Directions3, MIA Notify Directions4, MIA Notify Directions5, MIA Notify Phone
Education	Lists the educational qualifications of a Marine. Standard UDMIPS report used by small unit leaders around the Marine Corps. This report includes: -Formal civilian education -List of degrees earned -College courses taken -Military schools attended -MCI correspondence courses	Report Header: Rank, FName, MI, LName SSN, Plt Code, Co Code, Present RUC, Present MCC, Reserve RUC, Reserve RUC, Service Code, AFADBD Tables (multiple rows of data): 1) Formal Education -Civilian Education Level -Civilian Education Certification -College Major -Graduation Year 2) College Course (Individual Course List) -Class Title -Class Location -Class Completion Date -Credit Hours earned -Class Grade 3) Military School Data -Service School -Status Code (Pass, Fail) -Year 4) MCI Correspondence Courses -Course ID -Course Name -Class Date -Status Code (Enrolled, Complete)
BTR	Basic Training Record for an individual Marine. Standard UDMIPS report used by small unit leaders around the Marine Corps.	FName, MI, LName, SSN, Plt Code, Co Code, RUC, Present MCC, PMOS, Rank, Present Grade, Reserve RUC, Plt Code, Co Code, Service Code, AFADBD, Helmet Size, PFT Score, PFT Date, PFT Class, BST Attempted, BST Performed, BST Score, BST Date, Current Pistol Qual Date, Current Pistol Score, Current Pistol Class, Expert Pistol Qual Qty, Current Rifle Qual Date, Current Rifle Score, Current Rifle Class, Expert Rifle Qual Qty,

Report Name	Description	Data Fields
		Prior Rifle Qual Date, Security Lecture Date, Leadership Training Level, Leadership Training Date, HIV-III Lecture Date, Swim Qual Code, Swim Qual Date, Gas Chamber Date, Gas Mask Size, Gas Mask Type HIV Tested Date, ASVAB Test Date ASVAB Version, ASVAB GT Score, ASVAB ME Score, ASVAB CL Score, ASVAB EL Score, ASQT Score
Awards	Lists complete history of awards and commendatory material for the individual Marine. Standard UDMIPS report used by small unit leaders around the Marine Corps.	Report Header: Rank, FName, MI, Lname, SSN, Present Grade, Plt Code, Co Code, Present RUC, Present MCC, Reserve MCC, Reserve RUC, AFADBD, Service Code Table: Award Effective Date, Award Description Award Image (optional)
Security	Provides pertinent security and clearance information for an individual Marine	Rank, FName, MI, Lname, SSN, Present Grade, Plt Code, Co Code, Present RUC, Present MCC, Reserve MCC, Reserve RUC, AFADBD, Service Code, Component Code, Place of Birth, Record Status, Security Request Type, Clearance Award Date, Current Clearance Held, Clearance Eligibility, Security Investigation Agency, Clearance Completion Date, Security Investigation Type, Intelligence Training Hours
Pay and Leave Summary	Provides highlights of the LES. Used by Marine leaders to make decisions like whether to grant a leave request, obtain indications of pay problems, etc.	Rank, FName, MI, LName, Present RUC, Present MCC, PEBD, PMOS, Forecast Pay First Pay Period of Month, Forecast Pay Second Pay Period of Month, Actual Pay Amount, Pay Date, Beginning Leave Balance, Leave Earned, Current Leave Balance

Report Name	Description	Data Fields
LES	Leave and Earnings Statement. This DOD standardized report is the equivalent of the monthly "pay stub" found in civilian industry. Access to this report is through DFAS IT Systems. Detailing this report is beyond the scope of this thesis.	NA
Aggregate Personnel Rosters		
Personnel Roster	List of personnel assigned to the Company. This report must be configured such that it can be sorted in several formats: 1) Alpha Roster – alphabetized 2) By Rank – Highest to lowest 3) Group By Platoon Code; By Rank with Platoon 4) By DCTB - Oldest to most recent	Rank, FName, MI, LName SSN, Plt Code, Co Code, Sex, PMOS, DOB, EAS, DCTB, PEBD, DOR
EAS Roster	A roster with same data as Alpha roster but is sorted by EAS. EAS or End-of-Active-Service Date is the end date of the Marine's current enlistment contract. Important for Career counseling and personnel tracking	Rank, FName, MI, LName SSN, Plt Code, Co Code, Sex, PMOS, DOB, EAS, DCTB, PEBD, DOR
MOS Roster	A roster with same data as Alpha roster but is sorted such that Marines with same MOS are grouped together, then highest rank to lowest within the group.	Rank, FName, MI, LName SSN, Plt Code, Co Code, Sex, PMOS, DOB, EAS, DCTB, PEBD, DOR
Recall Roster	A roster of local address and phone number contact information. Usually sorted by last name. Also indicates if Marine is married and name of spouse if applicable.	Rank, FName, MI, LName SSN, Plt Code, Co Code, Local Street Address, Local City, Local State, Local Zip, Local Phone Married(M/S), Spouse FName
RED Roster	Record of Emergency Data. NOK or Next of Kin of Marines in the unit. Notify is the person to notify if a Marine is injured, missing in action, or deceased.	Rank, FName, MI, LName SSN, Plt Code, Co Code, Married (M or S) Spouse First Name, Spouse Last Name, Spouse Street Address, Spouse City, Spouse State, Spouse Zip, NOK Notify Code (Y or N) NOK Relationship, NOK Phone Notify Name, Notify Address
HOR Report	HOR or Home of Record listing for all Marines in the unit.	Rank, FName, MI, LName SSN, Plt Code, Co Code, HOR City

Report Name	Description	Data Fields
		HOR County HOR State
Deployment Roster	A roster used for manifests or uniform items request for deployments. Sorted by Rank or Last Name. Items in bold are not fields resident in MCTFS. Change to MCTFS schema required.	Rank, FName, MI, LName SSN, Plt Code, Co Code, EAS, Blood Type, Weight, **Boot Size,** **Cammy Trouser Size,** **Cammy Blouse Size,** **Cammy Cover Size,** Cap Size, Gas Mask Size, Married (M/S)
Security Roster	Provides pertinent security and clearance information for the unit.	Rank, FName, MI, Lname, SSN, Plt Code, Co Code, Place of Birth, Record Status, Clearance Award Date, Current Clearance Held, Clearance Completion Date, Security Investigation Type, Security Lecture Date, Intelligence Training Hours
Enlisted Pros and Cons	Provides a list of all junior enlisted Marines (E-1 through E-4) in the unit and their proficiency and conduct evaluation marks.	Rank, FName, MI, LName SSN, Plt Code, Co Code, MOS, EAS, DOR, Current Pro Mark Current Con Mark Current P/C Date Current P/C Occasion Avg Pro in Grade Avg Pro in Service Avg Con in Grade Avg Con in Service
COMRATS Report	Provides a list of Marines in the unit and their pay status with regards to what type of commuted rations (COMRATS) they received. COMRATS is the non-taxable allowance for subsistence (food) received by enlisted personnel. There are various rates for COMRATS and it is important for a Marine leader to track what rate a Marine receives.	Rank, FName, MI, LName SSN, Plt Code, Co Code, Local Street Address, Local City, Local State, Local Zip, Local Phone Married (M/S), COMRATS Type, COMRATS Amt
Aggregate Unit Training Rosters		
PFT Report	Physical Fitness Test Roster. Items in bold are not fields resident in MCTFS. Change to MCTFS schema required	Rank, FName, MI, LName SSN, Plt Code, Co Code, Gender, **Height**, **Weight**, Body Fat Pect, PFT Score,

Report Name	Description	Data Fields
		PFT Date, PFT Class
Gas Chamber	List all Marines in the unit and their current gas chamber attendance date and gas mask data.	Rank, FName, MI, LName SSN, Plt Code, Co Code, Gas Chamber Date, Gas Mask Size, Gas Mask Type
Rifle Score	Annual Rifle Range Qualification Roster for grades E-1 through E-6, O-1 through O-3, W1, and W2.	Rank, FName, MI, LName SSN, Plt Code, Co Code, Rifle Score, Rifle Class, Rifle Date
Pistol Score	Annual Pistol Range Qualification. Roster for grades E-6 through E-8, O-1 through O-6, and W1 through W4.	Rank, FName, MI, LName SSN, Plt Code, Co Code, Pistol Score, Pistol Class, Pistol Date
Swim Qual	Swim Qualification Roster. List all Marine in the unit and their current swim qualification data.	Rank, FName, MI, LName SSN, Plt Code, Co Code, WaterSurvCode WaterSurvDate
BST Roster	BST or Basic Skills Test Roster. Lists all junior Marines (E-1 through E-4) and their BST testing data.	Rank, FName, MI, LName SSN, Plt Code, Co Code, BST Attempted, BST Performed, BST Score, BST Date
DIC Report	Driver Improvement Course Training roster. DIC training required for all personnel less than 26 years of age. List only Marines under age 26.	Rank, FName, MI, LName SSN, Plt Code, Co Code, DOB, Age, DICQual (Y/N)
Weight and Military Appearance	Report lists all Marines presently assigned to the Weight Control and Military Appearance Program.	Rank, FName, MI, LName SSN, Plt Code, Co Code, Date Assigned Ht Wt Body Fat Qty of Occasions Assigned
Leadership Lecture Roster	Lists all Marines in the unit and their Leadership training status.	Rank, FName, MI, LName SSN, Plt Code, Co Code, Leadership Training Level, Leadership Training Date
HIV Roster	Lists all Marine in the unit and their last HIV test date and lecture training.	Rank, FName, MI, LName SSN, Plt Code, Co Code, HIV-III Lecture Date, HIV Tested Date
Security Lecture Roster	See Security Roster under Aggregate Personnel Roster Reports.	NA
Unit Statistical and Reporting Data		
Morning Report	Manpower Statistical Summary of all Marines Assigned to the Unit.	<u>Header Information:</u> Unit Name (Company) Company Cmdr Name & Rank

Report Name	Description	Data Fields
	Notes: 1. Used as a daily report or snapshot of the manning and status of a unit. 2. There are dozens of duty status codes in MCTFS. This report shows the most common duty statuses. For the obscure statuses, they shall still be reflected as such in MCTFS, but grouped together in the "other" category here with appropriate remarks. 3. Report consists of two parts: a statistical summary and a roster of Marines not in a "Full Duty" status. 4. This function may not be fully implementable at the company level due the segmentation of authority to enter different duty status codes. See discussion later in this chapter.	Date of Report Total Qty of Marines Assigned Unit Manpower Statistics Total Assigned Total Causality Total Hospitalized Total Sick-In-Qtrs Total Convalescence Leave Total UA Total Deserter Total IHCA Total Confined Total Other Legal Total Leave Total TAD Total FAP Total Other Total Full Duty Roster of Marines (Not Full Duty): Rank FName MI LName SSN Plt Code Duty Status Code Duty Status Description Remarks
Rank & MOS Status	Report for a statistical summary of MOS and Ranks in the unit.	Header Information: Unit Name (Company) Company Cmdr Name & Rank Date of Report Total Qty of Marines Assigned Tables: MOS & Qty Rank & Qty MOS and breakout of Rank within each MOS
Promotion Roster	This report lists all junior enlisted Marines (E-1 through E4) and their eligibility status for promotion to the next rank. Used for monthly recommendations for promotions. Group Marines by current Grade alphabetically. Leave blank columns for Recommendation and Remarks/Reason.	Header Information: Unit Name (Company) Company Cmdr Name & Rank Date of Report Table: Rank, FName, MI, LName SSN, Plt Code, Co Code, MOS, EAS, Eligible, Cutting Score
Training Stats	Statistical Summary of major training requirements for Marines in the Unit. Notes:	Header Information: Unit Name (Company) Company Cmdr Name & Rank Date of Report

Report Name	Description	Data Fields
	1. Different training events are required over different periods of time. The stats for that event reflect only data for the period of time in question. For example, PFTs are required during two semi-annual period each calendar year (Jan-Jun and Jul-Dec), whereas Rifle Qual is required each Fiscal Year (FY) which runs 1 Oct – 30 Sep. 2. Different training events have different requirements for individuals being required to undergo the training. For example, a Marine who is "WSQ" swim qualified need not ever undergo swim Qual during their career. Whereas a Marine who is a 4th Class swimmer must under swim Qual every FY. The training requirements for each event are not listed here but will be incorporated into queries. 3. This report may be developed as a single web page or several…one for each category.	Total Qty of Marines Assigned <u>PFT (Semi-Annual Period)</u> Semi-annual Period % Complete (Passed /Total) Qty Required Qty Medical Qty Passed Qty Required & not taken Qty 3rd Class Qty 2nd Class Qty 1st Class Qty of Max Score (300) Avg Score <u>Rifle Qual (FY)</u> % Complete (Qty Qual/Total) Qty Required (By Last Qual Date) Qty Qual during FY Qty UNQ Qty Marksmen Qty Sharpshooter Qty Expert Highest Score Name/Rank of Highest Score <u>Pistol Qual (FY):</u> % Complete (Qty Qual/Total) Qty Required (By Last Qual Date) Qty Qual during FY Qty UNQ Qty Marksmen Qty Sharpshooter Qty Expert Highest Score Name/Rank of Highest Score <u>Swim Qual (FY):</u> % Complete (QtyQual/Total) Qty Required Qty Qual during FY Qty UNQ Qty 4th Class Qty 3rd Class Qty 2nd Class Qty 1st Class Qty WSQ Qty MCWSI Qty Medical Waiver Name/Rank of MCWSI personnel <u>Gas Chamber Attendance (FY):</u> % Complete (Attended/Total) Qty Required (By Last Qual Date) Qty Attended

Report Name	Description	Data Fields
		Qty not Attended BST Testing (FY): % Complete (Passed/Total) Qty Required Qty Taken/Passed Qty Taken & Failed Qty Required not Taken DIC Training: % Complete (Passed/Total) Qty Required & Not Taken Qty Taken Wt Control: Qty Assigned Weight Control

Table 8. Reports Functional Requirements for Echelon II

E. DATA ENTRY FUNCTIONAL REQUIREMENTS

The task of data updates or data entry at the small unit level is a paradigm shift for the current generation of Marines. Several senior enlisted Marines have mentioned that many of the administrative functions now consolidated in the administrative unit were once found at the company level. Since the implementation of MCTFS in the 1960s and 1970s, all of these functions have been solely the domain of the trained administrator. With the development of TFAS, we will now see a shift of administrative functions back to the company level. For this concept to work, however, the Marine Corps must decide what data entry tasks are appropriate for the company to handle. As mentioned previously, we envision this part of TFAS development as potentially the most contentious as we move towards implementation.

1. Trust and Verification

In the Table 9 below, a list of data entry tasks we believe appropriate to the company is presented. Under the current system, all of this data is generated at the company level. The data is documented in hard copy format and then passed to various battalion level entitles (S-3 receives training data, Admin receives Pros/Cons, etc.) for review. At the battalion level multiple human hands touch the data and merely rubber stamp the company commander's submission. Only rarely is the company commander's data submission questioned. The process of multiple individuals at the battalion level

manually handling these documents is the cause of much of the current administrative system's unresponsiveness. The fact that all of the companies in a battalion funnel this data through these individuals, which is then entered by a few admin clerks, is also the cause for many of the data entry errors. The overworked diary clerk entering the data has no personal knowledge of the Marines that the data describes, nor does the clerk have any sense of personal ownership of the data. The clerk is only concerned with volume. Moving these functions to the company level and allowing them to be entered at the company level without review by battalion personnel is the right solution to achieve the envisioned TFAS efficiencies.

Critics of this proposal may contend that without the data being reviewed by battalion personnel, there will be a greater opportunity for fraud. In some ways they are correct. This point of view, however, is a negative one. We prefer to believe that the officers and staff non-commissioned officers found in the companies around the Marine Corps are just as trustworthy as those found in the battalion staff. We must trust these dedicated leaders in the companies to do the right thing. As a safeguard to this premise, we propose that we use information technology to our advantage and empower the battalion staff with a new type of oversight capability. Let the companies be the single point of entry for this data and allow immediate unit diary processing. The battalion staff currently has access to MCTFS data via UDMIPS and, hopefully, in the future a TFAS, Echelon III web site. Using these tools, the battalion staff may review the data entered at the company level. Special queries could be developed for this battalion oversight function to ferret out abuse and fraud. Leaders at the company level would understand that the oversight that was once conducted manually is now being conducted electronically and would be very reluctant to abuse their newfound authority. The bottom line is we must empower the leaders at the company level if TFAS is to have any real effect in revolutionizing our administrative processes.

2. Duty Status and Morning Reports

Beyond the issue of trust discussed in the previous paragraph, there are some data entry tasks that have a built-in conflict of interest. Currently, all companies generate a daily "morning report" at their level to report to their parent battalion the manpower status of their unit. These reports are generated through some separate and redundant

application from that of the MCTFS system/data. Morning reports around the Marine Corps are created in Word, Excel, Access, and other desktop applications. These reports are then submitted to battalion for consolidation. Prior to the removal of the administrative unit at the battalion level and their consolidation into a MSC (Division, FSSG, and Wing) level asset, morning reports were reviewed by the Personnel Officer or PersO (Officer in charge of the battalion administrative unit). Each report was several pages in length, and the "good" PersOs actually reviewed them and ensured the correct duty status codes were entered into that day's unit diaries. In reality, this process was rarely carried out on a daily basis. Again, the norm was that little scrutinization occurred at the battalion level. Battalion would simply compile the separate company reports, add the totals for the battalion and deliver the consolidated hard copy report to the battalion executive officer. With the admin consolidation, there is little doubt that there is even less review of these documents at the battalion level, and, certainly, a Marine's daily status is not being entered into and reflected in MCTFS.

The MCTFS system has the capability to accurately reflect on a daily basis the duty status of every single Marine in the Marine Corps, but the data values for duty statuses resident in MCTFS is highly suspect. If the duty status codes were actually accurate, leaders at every level of the Marine Corps could use the MCTFS data to get an accurate picture of a unit's readiness and deployability. The inability to get the right answer on a unit's deployability has plagued the Marine Corps for years. The cause is that no process has existed to allow the decentralized data entry of duty status codes for individual Marines by the small unit leaders who actually are familiar with these Marines. TFAS would crack this nut! The issue will be, 'will policy makers allow the company leadership to enter duty status codes into MCTFS?' This is not a simple question, since some of these codes have very negative meanings and possibly programmatic cascading effects in the MCTFS database that can affect such things as a Marine's pay and eligibility for promotion. For example, there are duty status codes for "Confined" when a Marine is incarcerated in a Marine brig, or "UA" when a Marine fails to report for work and has no authorization to be missing. These and other statuses affect pay and other manpower functions, e.g., when a Marine goes "UA", his/her pay is stopped until they return to duty. One can see that there is a conflict between merely reporting a duty status

code and its intended and practical effects in the manpower and pay system. Until now, this entire process has been the closely guarded function of the admin unit. I propose that we trust and empower the company to enter all types of duty status codes. I believe that 99% of the leaders at the company level will do the right thing. The benefit will be dramatically improved accuracy in the duty statuses data for the entire Marine Corps and accurate aggregate readiness figures. However, a more detailed analysis by trained administrators, pay personnel, and MCTFS programmers must analyze the current system for the second order programmatic effects of negative duty statuses. My recommendation is that there be a decoupling of any programmatic linkage between the duty status codes and automatic pay stoppages or other dramatic effects. Any negative duty status code entered at the company level should be reflected in a daily report reviewed by trained administrators at Echelon IV (the PAC), where action would then be taken to verify the status with the owning unit and to take the appropriate action (i.e. stop pay, etc.).

3. Company Data Entry Tasks

Table 9 below lists the data entry functions by data entry task name, description, and data entry fields. Omitted from the table are the data fields needed by the TFAS user in order to have a "data context" for the data entry. The data context for data entry would be those data fields needed to be displayed to the user but are not editable. For example, for the data entry task of entering a PFT score and PFT date for an individual Marine, the user must be able to view, but not edit data on this Marine. This data would include:

- SSN
- Lname
- Fname
- MI
- RUC (Reporting Unit Code) for parent battalion
- Company
- Platoon

All of the data entry tasks listed below are assumed to have these read-only data fields available as a context for the data entry task listed.

Data Entry Task	Description	Data Entry Fields
Personnel & Unit Information		
Morning Report	See discussion in Paragraph 2 above.	Duty Status Code Remarks
Platoon Codes	Allows company to manage the assignment the contents of this data field. Notes: 1. MCTFS programmers will need to add new data fields to the MCTFS schema. Currently, there is only one Plt Code for a Marine. There needs to be a Plt Code field for each RUC associated with a Marine (e.g. Present RUC, TAD RUC, etc.) 2. Currently, the values for any typical unit are not uniform and make queries by Plt code impossible. This affects the ability to present accurate "Platoon Vie ws".	Platoon Code
Promotion Recommendation	Promotion recommendations occur on a monthly basis. Presently, this process involves a manual print out of eligible Marines (E-1 through E-4) by Admin. Companies then annotate the report with Yes or No for promotion recommendation. Admin then enters these Yes/No values into MCTFS.	Pro Recommend Flag (Y/N)
Pros and Cons	Junior Enlisted Marines, grade E-1 through E-4, are evaluated by the assignment of Proficiency and Conduct marks (Values: 0.0 through 5.0; increments of 0.1). Various occasions for assignment: semi-annual, transfer, promotion, etc. Company commander always assigns these codes.	Proficiency Mark Conduct Mark Occasion Date
Equipment Sizes	Equipment and uniform sizes for individual Marines. Accurate data in MCTFS for these items can be used by local logistics agencies to procure the right amounts of equipment for deployments, etc.	Helmet Size Cap Size Cover Size Blouse Size Trouser Size Boot Size Gas Mask Size Gas Mask Type
Recall Data	The comp any typically accurately maintains this data redundantly. Allow company to enter this data and have a single accurate source of recall data for all Marines.	Local Street Address, Local City, Local State, Local Zip, Local Phone Work Phone
RED Data	Record of Emergency Data. 1. Use for notification of a Marine's family members in the event a Marine is injured, MIA, or killed on active duty.	Spouse First Name, Spouse Last Name, Spouse Street Address, Spouse City, Spouse State, Spouse Zip,

Data Entry Task	Description	Data Entry Fields
	2. Some training of company office personnel on how to instruct Marine to verify the RED data will be required. 3. Requires a hard copy be signed/verified by the Marine and kept on file in company office.	Father Name, Father Address, Mother Name, Mother Address, NOK Notify Code, NOK Relationship, NOK Phone Notify Name, Notify Address, Notify Directions1, Notify Directions2, Notify Directions3, Notify Directions4, Notify Directions5, MIA Notify Name, MIA Notify 1st Phone, MIA Notify 2nd Phone, MIA Notify Relationship, MIA Notify Directions1, MIA Notify Directions2, MIA Notify Directions3, MIA Notify Directions4, MIA Notify Directions5, MIA Notify Phone
Training		
PFT Score	Physical Fitness Test Score. PFTs are typically administered at company level. Allow data entry by the unit that conducts the PFT. (Vision: PFT score entered during and at the site of the PFT via wireless PDA) Option: Marines who fail the PFT or who are overweight and must be assigned to weight control must be screened by battalion S-3 training personnel. Negative PFT and Weight Control data entry relegated to Echelon III (battalion). Company can only enter passing PFT Scores and medical waivers.	PFT Score PFT Date Height Weight Body Fat
Gas Chamber	Gas Chamber attendance date data entry	Gas Chamber Date
DIC Training	Drivers Improvement Course attendance data entry.	DIC Training Flag (Y/N)
Leadership Lecture	Leadership Lecture attendance date data entry.	Leadership Lecture Date
HIV Lecture	HIV Lecture attendance date data entry.	HIC Lecture Date
Security Lecture	Security Lecture attendance date data entry.	Security Lecture Date
BST Testing	Basic Skills Test data. Notes: 1. A written and oral examination administered to junior enlisted Marines annually to test their proficiency of basic military skills and knowledge.	BST Attempted, BST Performed, BST Score, BST Date

Data Entry Task	Description	Data Entry Fields
	2. The level of command (company or battalion) at which the test is administered varies by unit. 3. Recommendation: Data entry possible at Echelon II and III.	

Table 9. Data Entry Functional Requirements for Echelon II

F. MULTI-ECHELON DATA TASKS

As the other echelons of TFAS are developed, there will be other data entry functions added. These functions, however, will be part of a chain of events involving multiple leaders across echelons, over a period of time. The final MCTFS data entry (creation and submission of a unit diary) will most probably not occur at Echelon II, but at higher levels. The role of the TFAS, Echelon II web site in this functionality will be to display web pages of manpower data requests in process or in some intermediate state. The vision for this functionality is that data is transmitted to a web/database server (emails or web forms containing admin requests, e.g. a leave request) by one TFAS User where it can be processed and stored persistently in a database. Then, when another TFAS User in the chain of command of the first user accesses the TFAS web site for his/her unit, data is displayed regarding the request. For example, the second user would view a list of all leave requests submitted by Marines in the unit under his/her command and this user must approve or disapprove them. Approving a leave request initiates further action similar to the interaction just described. This process continues among multiple users across echelons over time until the data task is completed and a unit diary entry can be run to reflect the action (e.g., charge a Marine for leave days taken after Marine returns from leave). This type of multi-echelon interaction will further empower the company and reduce the workload in the PAC. However, this functionality cannot be implemented without a total system approach to the design effort of TFAS.

With a detailed description of the functional requirements of the TFAS, Echelon II web site prototype, we have the basic building blocks necessary to construct the prototype. In the next chapter, we will describe the system architecture for the TFAS, Echelon II web site prototype and it's integration with MCTFS.

V. WEB SITE ARCHITECTURE AND INTEGRATION

Armed with an understanding of the current manpower data system, MCTFS, the broad functional requirements of TFAS, and the detailed functional requirements of the TFAS - Echelon II web site, it is now possible to describe the web site architecture for the prototype. The purpose of this chapter is to outline the system architecture for the Echelon II web site prototype and its integration with MCTFS. The architecture will be described both in general terms, which can be used as a template for any implementation regardless of vendor specific software, and in specific terms. The specific architecture presented will detail the setup and integration of Microsoft products used in our prototype.

A. GENERAL DESCRIPTION

1. Distributed Architecture

The conceptual architecture for TFAS presented in Chapter 3 served as the basis for our design. The conceptual architecture gives us a good idea of how data will flow, but does not address how the system would look as a whole. From a total systems perspective, we have two choices: a single web and database server that provides service to all clients throughout the entire Marine Corps, or a distributed network of web and database servers. Presently, the TFAS planners are modeling their system architecture on the single "national" web and database server model. This approach is developing due to the use of the Marine OnLine (MOL) web site as the prototype for Echelon I. MOL runs on a single server in Quantico, VA and receives data from a mirrored copy of the ODSE. The single server approach is appropriate for several reasons.

- Ease of software updates on a single server
- Streamlined administration of a single web site
- Simple management of access and user accounts

While the single server model offers the benefits of tighter control by a single entity, it may not be the best approach for the full implementation of TFAS. Presently, MOL only services individual Marines viewing their individual data. The worst-case load on that server is the infrequent traffic of individual Marines viewing read-only reports. When all

Echelons of TFAS are deployed, the load on that single server will grow exponentially. From Chapter 2, we learned that there are over 3 million data transactions (unit diaries) conducted yearly through the present UDMIPS applications in admin shops. Once data transactions are partitioned out to the TFAS echelons and Marines start using the TFAS web site, the amount of write transactions may very well grow by a factor of five to ten. Justification for this prediction is easily derived from the fact that instead of 2000 trained administrators inputting data in UDMIPS, we will now have virtually the entire leadership of the Marine Corps submitting data on a daily basis. The number of users could easily exceed 50,000; encompassing every leader or unit staff member from Squad Leader up to Division Staff and PAC personnel. Additionally, the ability to view many different reports (read-only) at Echelons II – V will dramatically increase the load on the server. Even with parallel servers for load balancing and failover capability, there will be issues of throughput to the client and response time. By sending large amounts of data across the continental United States and overseas (bases in Okinawa and Hawaii, as well as deployed units), there could be a significant and noticeable slow response. Lastly, a single, tightly controlled TFAS web server will stifle innovation at the local level. For TFAS to be successful, Marines will have to want to use it. During the initial deployment of TFAS, users will certainly want to add functionality unique to their locale. A distributed architecture would allow for such innovation and tailoring to individual unit needs, while keeping core TFAS functionality intact. A single server approach would inhibit this innovation and tailoring to local user's needs.

Our recommendation for the overall system architecture is the distributed approach. This architecture consists of two components: (1) a system of distributed databases and servers synchronized with the MCTFS CMF, and (2) local TFAS web servers. In Figure 10 below, a macro view of this distributed architecture is depicted. In this figure, a server at the national level located in St. Louis, MO sends and receives data from the single source of data resident in MCTFS. Data in the form of unit diary files are transmitted to the national server from major Marine Corp bases where they are processed on the MCTFS main-frame. After the cycle, updated MCTFS data in the CMF is refreshed in the ODSE. After the ODSE is refreshed, data updates from the ODSE are passed back to the servers at the base level.

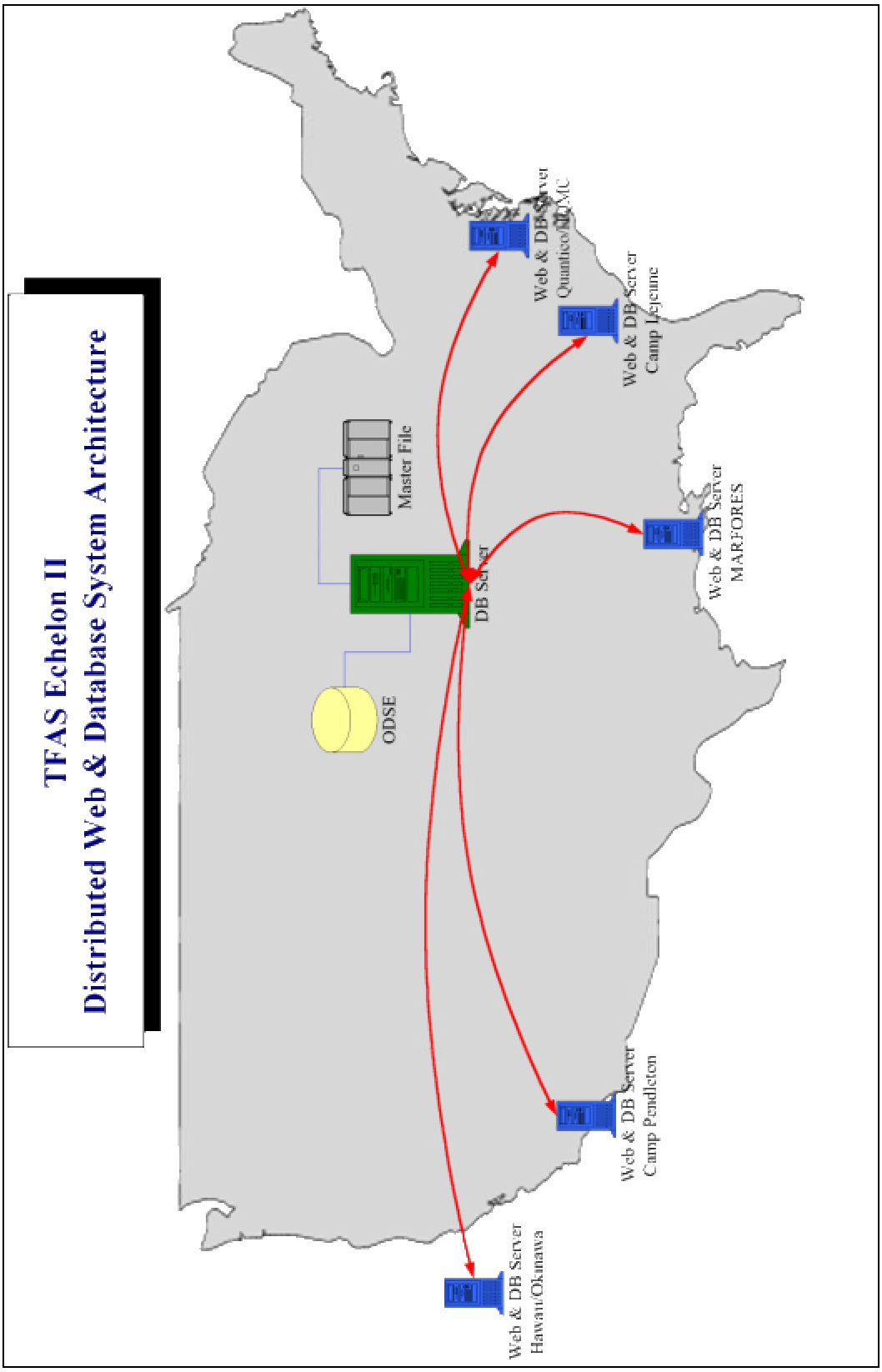


Figure 10. TFAS Echelon II Distributed Web and Database System Architecture

The servers at the base level are the focal point for all data transactions for local users. With a distributed architecture, the large volume of daily data transactions by the clients is contained within a small geographic area. This will serve to reduce the response time and the load on the web and database server that services client requests. Local users could add functionality unique to their needs by authoring their own web pages and loading them on the server. Core functions of the TFAS web site would not be altered by local users in order to ensure a uniform interface around the Marine Corps. Recommendations and prototype web pages for changes to core functions could be forwarded to the national level where manpower and software experts could review them. Approved recommendations could be released twice a year, paralleling the updates in MCTFS mainframe and UDMIPS application.

The drawbacks to the distributed approach center on the lack of central control and, potentially, increased system costs. Maintaining distributed servers increase hardware and software costs compared to a single national server. Facilities and trained web masters/database administrators would be needed to staff each location, incurring further costs for manning and infrastructure. Lastly, there will be increased complexity in systems management due to the technical challenge of keeping multiple local copies of the ODSE synchronized with the National ODSE. Even with these drawbacks, however, we believe the benefits of a distributed architecture outweigh the costs. However, before the TFAS program matures any further, a detailed study should be conducted to compare the benefits, drawbacks and costs of each approach.

2. TFAS Systems Architecture and Data Flow

The assumption for our prototype is that the Marine Corps will adopt a distributed architecture. As a result, we have defined the system components for the architecture at the local Marine Corps Base and its interface with MCTFS systems. In Figure 11 below, a model of the TFAS systems architecture is shown along with a description of the data flow through the system.

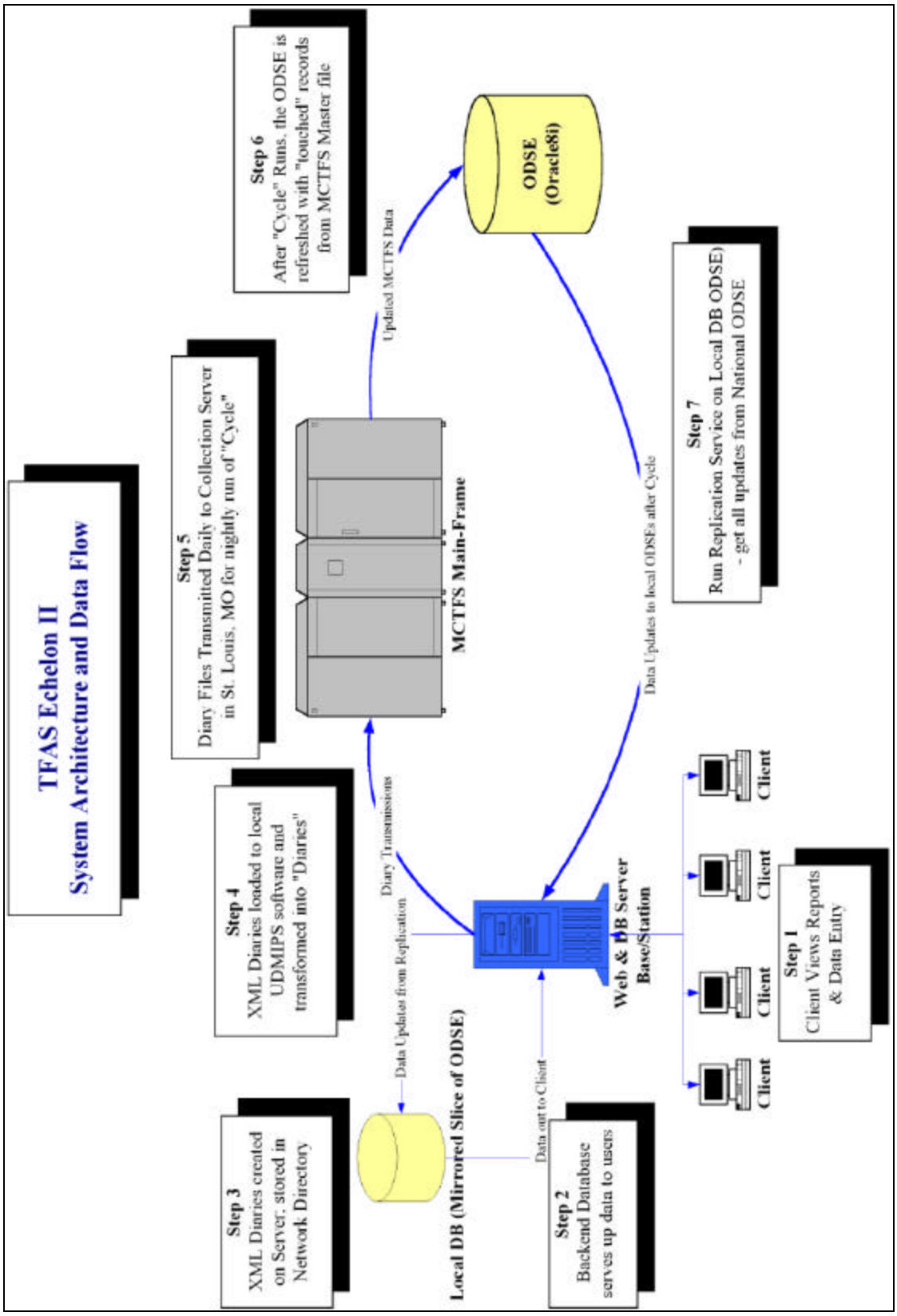


Figure 11. TFAS Echelon II System Architecture and Data Flow

When the reader compares this architecture to that of the conceptual model for Echelon II presented in Figure 7 of Chapter 3, it is now clear how data will actually flow through the system. The user requests web pages for a certain function. Data for the report or data entry task is read from the local backend database, which is a mirrored copy of the ODSE, and returned to the user. If the user is entering data, the transaction is returned to the server where the web script processes it. The output of the script is the creation of a well-formed XML file that is the XML implementation of a unit diary. These XML diaries are stored in a network directory until they are processed for transmission. At the end of the business day, the XML diaries are converted into "normal" unit diaries through the UDMIPS generic import utility and then are transmitted to the diary collection server in St. Louis, MO. At this point, the current manpower data process is the same. All of the day's unit diaries are processed in MCTFS. After the MCTFS processing cycle, the ODSE is refreshed with any "touched" data (records that were updated during the cycle). Once the ODSE is refreshed, the local ODSE databases at the bases can be updated through automated replication services. Now the data cycle is complete and all data updates entered via a web page are now reflected in MCTFS, the ODSE, and the local ODSE.

The system architecture shown in Figure 11 is specific enough to show how the system would work, yet general enough so that any specific web server and database software could be employed in the final implementation of TFAS. This model is our template for a specific software implementation of the TFAS, Echelon II web site prototype. The remainder of this chapter will focus on the characteristics of specific software products used in our prototype to implement the web site at the local level. The description of this architecture at the local level will consist of three components: network environment, the web server, and the backend database.

B. NETWORK ENVIRONMENT

1. General Description

As late as 1998, desktop computers in the Marine Corps were a collection of unconnected, aging machines configured in stand-alone mode, or as isolated workgroups. Many desktop computers were still running the Windows 3.1 operating system on

outdated hardware configurations (286 Processors with no networking interface cards for LAN connectivity). Internet access was extremely limited and email capability rarely existed below the battalion level. Over the past five years, a significant transformation has occurred aboard Marine Corps installations. During this time, virtually all of the desktop workstations were upgraded to Pentium class computers with LAN capability. Many bases have now been wired with CAT 5 LAN wiring and LAN hubs. The operating system on almost all desktops is now either Windows NT 4.0 or Windows 2000 Professional. Units have been staffed with computer technicians and network administrators. Internet access and email capbility is almost universal and is available to leaders at all levels. With the advent of NMCI and the ten-year contract with EDS, the integration of the Marine Corps' computer networks will become universal and systems will be upgraded in a timely manner. This is a necessary condition for the success of the TFAS, Echelon II system.

In designing the implementation of the TFAS, Echelon II web site prototype, the network environment is a critical consideration. Even though the Marine Corps's networks are not yet universally integrated across the entire Marine Corps, there are integrated networks aboard major bases and stations. Specifically, this local network integration means that the network is organized as a single domain or a collection of trusted domains. This domain structure allows for the seamless management of user accounts for network access and email capability. This means that once a Marine leader at a particular base is granted a network account, the Marine can log on to any network desktop aboard that base and be recognized. This universal access to network resources through a single network account is the linchpin in our implementation of the prototype. Specifically, a TFAS user will be authenticated and allowed only those data functions associated with their network user account. This use of network user accounts within the local network aboard a base will be critical in the implementation of TFAS user access permissions associated with "Roles" defined in Chapter 4. It will also allow for manageable administration of local TFAS user accounts.

2. Windows NT/2000 Operating System

Based on the preceding description of the network environment available in most Marine Corps units, the foundation of the prototype's design is modeled around the

existence of Windows NT/2000 user accounts and a system of trusted domains. At the time that the prototype was developed, no networked computer was available at NPS to set up a simulated network environment. The computer used for the prototype was my home computer, running the Windows 2000 Professional operating system. This operating system is designed to be a client/work station and is not capable of implementing domains and Active Directory Services. In order to simulate a network of TFAS users similar to that found aboard a Marine Corps base, a system of computers would need to be set up with one or more of these computers would serving as a domain controller. These domain controllers would need to run the Windows NT Server, Windows 2000 Server or Windows 2000 Advanced Server operating systems. Each of the domain controllers would maintain a list of user accounts for their domain (collection of workstations belonging to the specified domain). The list of users maintained by the domain controller is the Active Directory. The collection of domain controllers could then be linked such that one domain controller "trusts" the user accounts listed in the Active Directory of another domain controller. In this manner, all user accounts defined within each domain aboard a base or geographical area would be recognized universally.

While simulation of the actual network environment was not possible for the prototype, it was nevertheless possible to demonstrate the concept using the Windows 2000 Professional computer. A small group of computers running Windows 2000 Professional can be linked together (LAN connection) and be part of a networking workgroup. In a workgroup, each computer maintains its own list of user accounts. If a user on one workstation wishes to access another, they must have a legal user account on that second machine. While this is a different implementation of user accounts than envisioned aboard a Marine Corps base, it can serve as a suitable network architecture in which to demonstrate the proof of concept for the prototype. In Figure 12 below, a screen shot of the Computer Management MMC (Microsoft Management Console) on the prototype computer is shown. Using this MMC, the network administrator maintains a list of authorized user accounts for the network. Further, a local group has been created called "TFASUsers." This local group is used to implement authorization/access to the various files associated with the web site prototype. As network users are granted authority to access the TFAS web site, their network user account is added to this group

in this MMC interface. Individual file (web page) permissions have the TFASUser group listed in its security/access permissions properties. Through this arrangement of network user accounts, groups, and file permissions, there will be a fine degree of control over all or parts of the web site and flexible daily management of TFAS user access. Lastly, the individual TFAS user will have a single "User Name" and password for both network access and TFAS web site access.

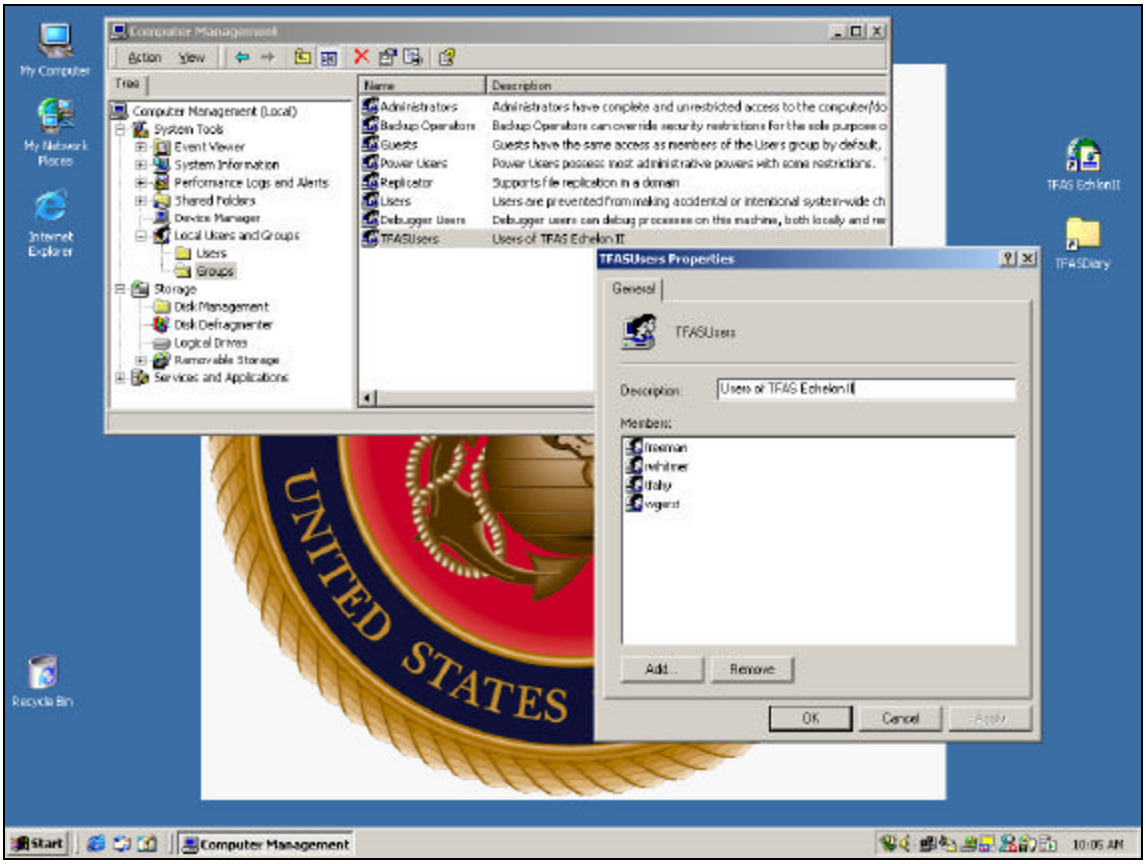


Figure 12. Network and TFAS User Account Management

C. THE WEB SERVER – IIS 5

While there are many web servers on the market that could be used on a Windows 2000 machine, the prototype will be implemented using the Windows 2000 integrated HTTP (Hypertext Transfer Protocol) server, Internet Information Server - 5 (IIS-5). "The advantage of IIS-5 over its competitors is its total integration with Windows 2000. IIS-5 integrates totally with the Windows 2000 Active Directory, security, and file access

structure." [Ref. 12] The use of IIS-5 eliminates the requirement to maintain a separate list of user accounts to control access to the web site. This and other features make it the best choice for use in our prototype.

IIS-5 is a software program that runs as a service on the Windows 2000 machine whose function is to deliver web content to network requests. This web content can be static HTML files or scripting content. The scripting languages supported by IIS-5 are JavaScript and Visual Basic Script (VBScript). The primary scripting format is Active Server Pages (ASPs) written in VBScript. The integrated ASP engine (software module) in IIS-5 allows web authors to write ASP scripts, which are like small computer programs that run on the server and deliver dynamic web content to clients. It is this dynamic web content capability that will be used almost exclusively in implementing the prototype and it will enable the tailoring of manpower data content to the identity of the requesting TFAS user.

Although every version of the Windows 2000 family of operating system comes with IIS-5 software, the available features of IIS-5 vary. With the prototype running on a Windows 2000 Professional machine, the system is limited to no more than ten connected users. [Ref. 12] Additionally, some of the more sophisticated features, like remote web site administration, are not available. However, for the purposes of developing the prototype, the "Personal Web Server" variant of IIS-5 running on our Windows 2000 Professional machine will suffice to demonstrate the technical proof of concept.

Beyond the integration of user accounts between IIS-5 and Windows 2000, IIS-5 has many easy-to-use features that make web site management less complicated. These features are presented below. Screen shots of the specific IIS-5 settings for the TFAS prototype are intermingled in the list to demonstrate our implementation.

- Intuitive web site management through a MMC (Microsoft Management Console) on the local machine. Through a series of menus, drop down lists, and dialogs, making changes to the many settings within a web site is made relatively easy by IIS-5. Many other commercial web server's settings must be adjusted by editing cryptic DOS-command line like text in several different files. In the Figures 13 and 14 below, the MMC for IIS-5 is shown for the prototype and its properties window. It is through this interface that settings are made to the web site, web directories, and individual web pages. A portion of the directory structure and content of the prototype can be viewed in the figure. A complete site map of the prototype is presented in Chapter 6.

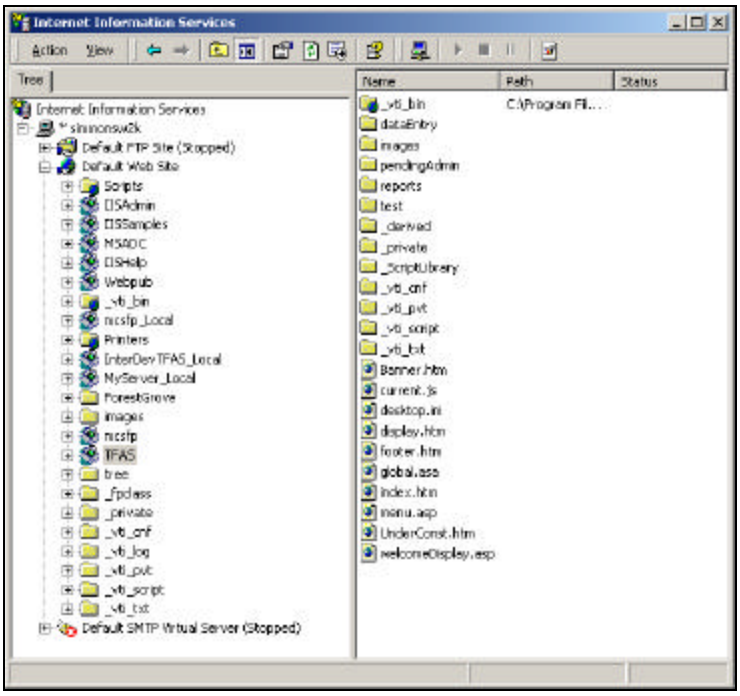


Figure 13. TFAS Web Site Management – IIS-5 MMC

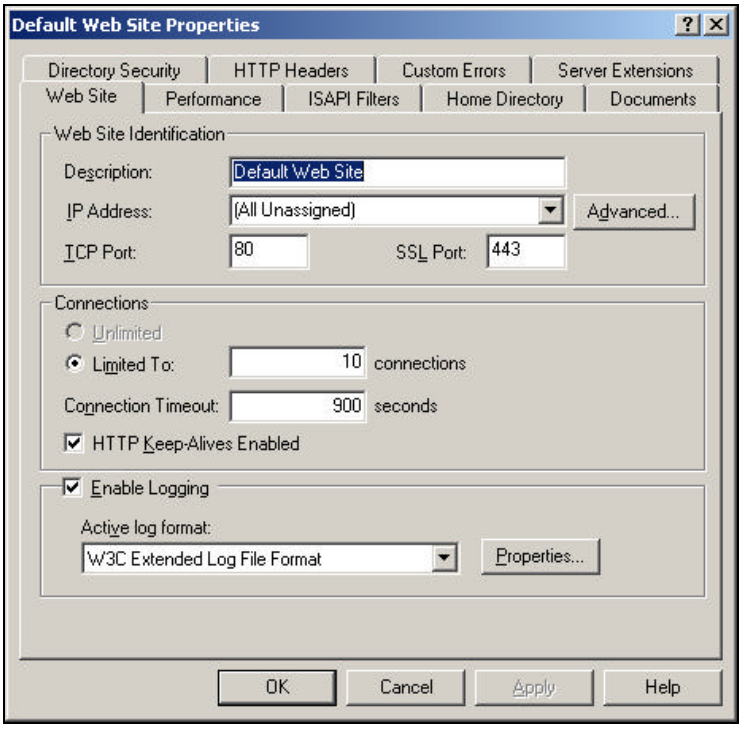


Figure 14. TFAS Web Site Properties - IIS-5

- Remote Management. Web site administration from a remote computer through an MMC or Web Page.
- Delegated Management. The web master can give web site administrator privileges to web subdirectories to the authors of the content in those subdirectories.
- Front Page Server Extensions. IIS-5 permits remote authoring through Microsoft Front Page. While Front Page Server extensions are part of the default features of IIS-5, the use of Front Page as a web authoring for TFAS development is limited. Front Page does have some database "wizards" that aid the web author to construct web pages populated with database data, but is limited in functionality. The level of sophisticated programming required in TFAS development requires a more powerful authoring tool. In Chapter 6, a discussion of our web-authoring tool, Microsoft InterDev 6.0, is presented. IIS-5 supports InterDev server extensions.
- Fine Degree of Access Permissions. Through the combination of Windows 2000 permissions and IIS-5 permissions, a variety of access permissions for the web site, directories, and individual files (web pages) can be assigned. Figure 15 below shows the settings for the TFAS top-level directory.

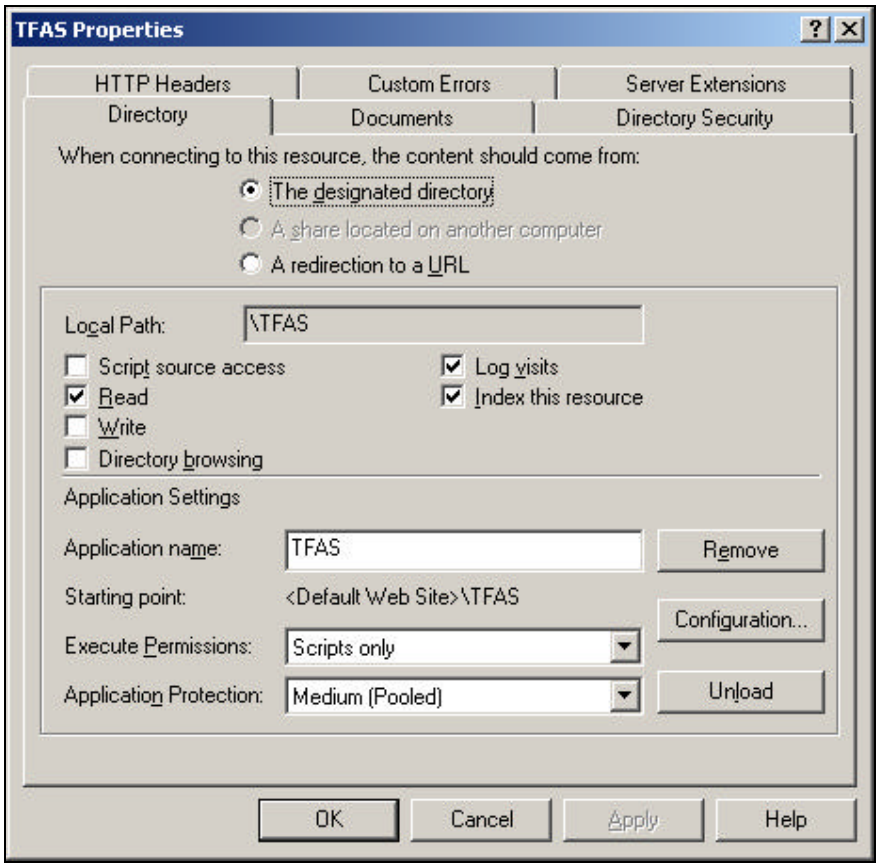


Figure 15. TFAS Web Directory Properties in IIS-5

- Encryption. IIS-5 fully supports the use of Secure Socket Layer (SSL) 3 and Public Key Cryptography, which provides a secure channel for communication between client and server. Easy to use Certificate Wizards are available to install certificates and enable SSL communication.
- Several User Authentication Modes. User authentication can be easily be configured for Anonymous, Basic, or Windows Integration Authentication. IP number or domain membership of the client versus a specific user account can also control access. The figure below depicts our setting for integrated Windows authentication.

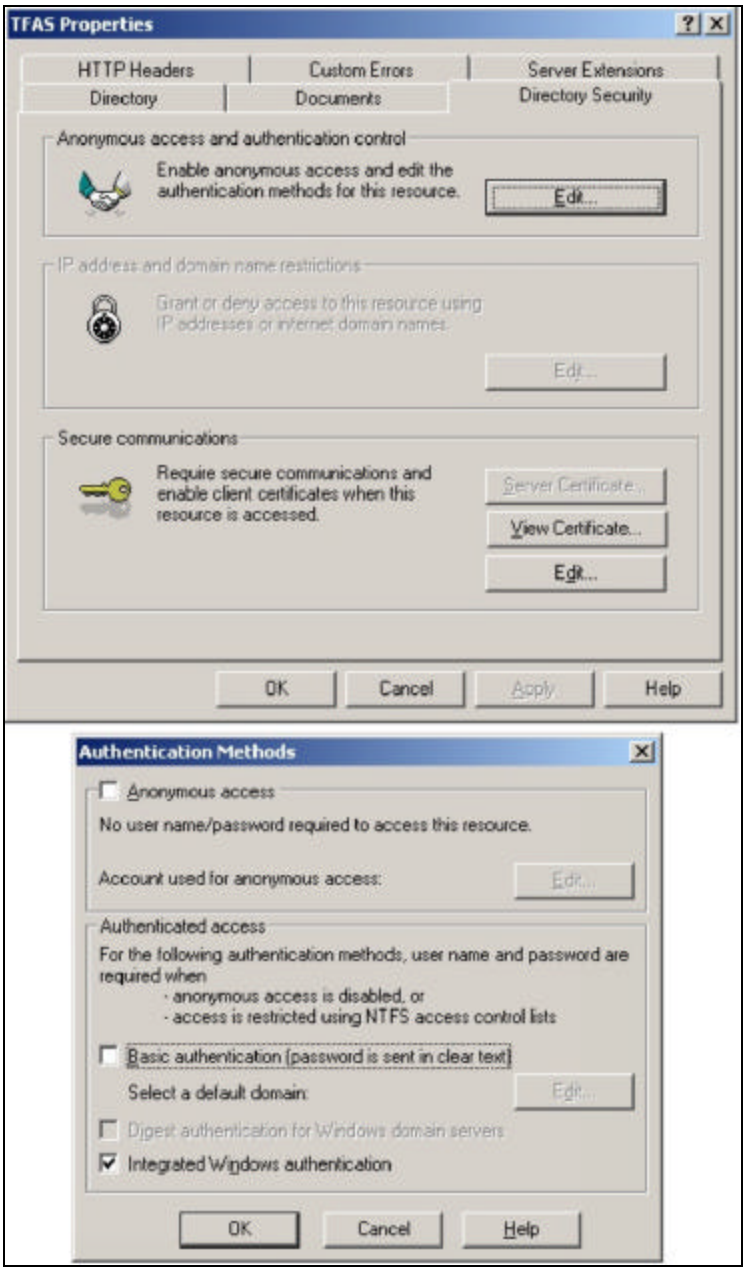


Figure 16. Integrated Windows Authentication - TFAS Web Site

- Customizable error codes and error response pages.
- FTP (File Transfer Protocol) Support. IIS-5 has its own built-in FTP server that allows web authors to transmit content to network/IIS-5 directories.
- Auditing and Logging Capabilities. Default logging generates daily text files of web transactions. Another logging feature allows for the auto-population of an Access relational database.
- Load Balancing and Clustering. This capability is available when multiple IIS-5 servers are linked together in order to balance server load and for failover.
- Multi-site Hosting. On one IIS-5 server multiple web sites can exist either with differing IP addresses, the same IP with different port numbers, or the same IP number with different host headers.
- Socket Pooling. The IIS-5 software manages the connections by pooling the sockets available, which increases server performance.
- Bandwidth Throttling. IIS-5 allows for the designation of allowable bandwidth for any particular application running on the server. Some applications may need more bandwidth to deliver larger amounts of content, whereas other applications may have minimal bandwidth requirements. [Ref. 13]

D. BACKEND DATABASE

The focus of the TFAS concept is the accessibility of MCTFS manpower data. The single source of manpower data resides on the MCTFS mainframe in a flat file, proprietary format and is inaccessible for web applications. However, as mentioned earlier, there is a relational database version of the data available via the Operational Data Store Enterprise (ODSE). The ODSE is synchronized with the updates to MCTFS data after each cycle, so the ODSE is a valid source of current data. The ODSE in St. Louis, MO is as closely guarded as the MCTFS mainframe. Accordingly, the personnel responsible for the ODSE at TSO in Kansas City, MO are reluctant to allow web servers to connect directly due to security and data integrity concerns. For this reason, mirrored copies of the ODSE already exist at several major Marine Corps installations, like the one at Quantico, VA, which is used as a data source for Marine OnLine, and various other software applications with Headquarters Marine Corps entities. This model of using a mirrored copy of the ODSE at the local Marine Corps base as the data source is the approach we have chosen for the implementation of our TFAS, Echelon II web site.

In order to program the various web pages for the prototype, a development copy of the ODSE was needed. The actual ODSE resident in St. Louis is an Oracle 8i

Enterprise relational database. At present, NPS does not offer any instruction on the use of commercial level Database Management Systems (DBMS) like Oracle, nor is any software available. Oracle systems are widely used by large organizations with national or global reach. Oracle software products are designed for large corporate clients and their programming require intensive training. The design of the database, queries, and other functions is strictly the domain of highly trained database administrators. The unavailability of Oracle software at NPS coupled with the steep learning curve required to become accomplished in the system led us to reject Oracle as the backend database for the prototype. Instead, we selected Microsoft SQL Server 2000 Standard Edition because it is a commercial grade DBMS like Oracle, it has tight integration with the other Microsoft system components (Windows 2000 and IIS-5), and it has a very intuitive interface for designing tables, queries, and stored procedures. During a thesis research trip to DFAS in Kansas City, MO, we obtained a partial copy of the ODSE in SQL Server 2000 data format.

1. Database Schema

The data contained in the national ODSE reflects all 1800+ fields of data for both manpower and pay information for each individual Marine found in the MCTFS master file. Additionally, the ODSE contains over 500,000 records for all active duty, reserve and retired Marines. As a result, the ODSE is quite a large data set (many gigabytes of data) and has hundreds of tables. To develop the prototype for our Echelon II web site, we did not require a complete copy of the ODSE, but only those tables that containing the data fields as defined in the functional requirements presented in Chapter 4. For these reasons, we obtained only a partial copy of the ODSE to use as our development backend database. This data was current as of 7 April 2002.

Our development backend database (the prototype's local ODSE) consists of 141 tables of which 24 tables contain data on personnel and 117 tables are "lookup" tables. Lookup tables are a common technique used in database design to ensure data integrity and to increase efficiency of the database. These lookup tables map MCTFS codes found in the 24 personnel data tables to longer, human-readable English versions for the meaning of the codes. Many of the data views will require "joins" of these personnel data tables to the lookup tables in order to present meaningful information to the

requesting TFAS user. Even in our partial version of the ODSE, there are still hundreds of fields. Enumerating them here would serve no purpose, but Table 10 below lists the 24 personnel data tables.

Table Name	Description
Individual Marine	Table with Primary key of SSN and many data fields common to all Marines.
Enlisted	Data associated only with enlisted personnel.
Officer	Data associated only with officers.
Training	Training data on personnel. PFT Scores, Rifle Range scores, etc.
Security	Security classification data on personnel.
Composite_Score_R123	Composite score data for junior enlisted Marines. Used to determine promotion eligibility.
Proficiency_Conduct_R110	Proficiency and Conduct Marks. Evaluation data on junior enlisted Marine's performance.
MCI_Course_Info_R120	Data associated with Marine Corps Institute correspondence course enrollment and courses completed by personnel.
Civilian_Educ_Info_R147	Civilian education data for personnel.
Govt_Eqp_Opr_License_R141	Licensing information for military equipment like HMMWVs, etc. for personnel.
School_Special_Skill_R136	Military Schools and skills data for personnel.
Test_Score	ASVAB and other military skills test data on personnel.
Off_Duty_Education_R121	Additional civilian education data.
Awards_R143	Awards History for personnel.
Marine_Not_Notify	Data on persons not to notify if a Marine is injured, MIA, or killed.
Marine_Info	Additional Individual Marine data fields.
Marine_Arrears	Death benefit data for personnel.
MOS_Additional	Military Occupational Skill data for personnel.
RED_MIA_Notify	Record of Emergency Data information for personnel.
Marine_Next_of_Kin	Next of Kin data for personnel.
Grade	Promotion related data for enlisted personnel.
Marine_Dependent_Summary	Dependent (immediate family – spouse, children) data.
Marine Family	Dependent (immediate family – spouse, children) data.
Marine_Child	Dependent (immediate family – spouse, children) data.
Marine_Command	Data related to units in the Marine Corps. This table does not contain personnel data and thus no foreign key reference to SSN. This table is similar to a lookup table, but contains data regarding unit information. Records in the Individual Marine have a foreign key reference to the unit Ids or RUCs found in this table. Has many records.

Table 10. Personnel Data Tables in the ODSE

Of these 24 personnel tables, the "Individual_Marine" table is the focal point of the data. The table's primary key is the social security number (SSN) of the service member and is the unique identifier for that record. The remaining 23 personnel data tables have a

primary key which is the SSN and, thus, have a foreign key reference to the SSN field in the Individual_Marine table.

One other critical table is the "ODSE_CYCLE_INFO" table. This table has only a few fields and maintains a history of the MCTFS cycle and the ODSE refresh process following the cycle. The two most important fields in this table are the "Cycle_Date" and "Extract_Date." These fields play an important role in keeping the local ODSE database synchronized with the "national" ODSE. During any manual or automated replication process, the dates in these both the local ODSE and national ODSE tables are compared in order to determine if a refresh should take place.

It should be noted that the schema of each table, field names, table names, and relationships in our SQL Server 2000 partial ODSE are identical to those found in the Oracle 8i national ODSE. We felt it wise not to alter any aspect of the existing structure of the database. The reason for this decision was to ensure simplicity when conducting replication. By keeping the same table structure and naming scheme as the national ODSE, no mapping of fields and tables would be required since the two relational databases would be identical. We did, however, add fields to existing tables and other tables to increase functionality. These additions will be described later in this chapter.

2. Microsoft SQL Server 2000

The selection of SQL Server 2000 as our backend database was made casually. As mentioned previously, it was selected mainly due to the ease of use compared to Oracle, as well as its tight integration with other Microsoft products. However, there are many features of Microsoft SQL Sever 2000 that make it desirable as a backend database not only for our prototype development, but also for the actual deployment of TFAS web sites and local ODSE databases. Salient features of SQL Sever 2000 are listed below commingled with screen shots of some of these features demonstrate our prototype's backend database configuration. As a commercial grade DBMS, SQL Server 2000 offers the following features. [Ref. 14]

- Database Management. In Figure 17 below, the SQL Server 2000 Enterprise Manager MMC is depicted. This interface provides an easy to navigate menu system for finding components of the database and adjusting their settings using menus and dialog boxes rather than complex command line instructions. There are also many "wizards" available for common tasks like creating tables, queries, and stored procedures.

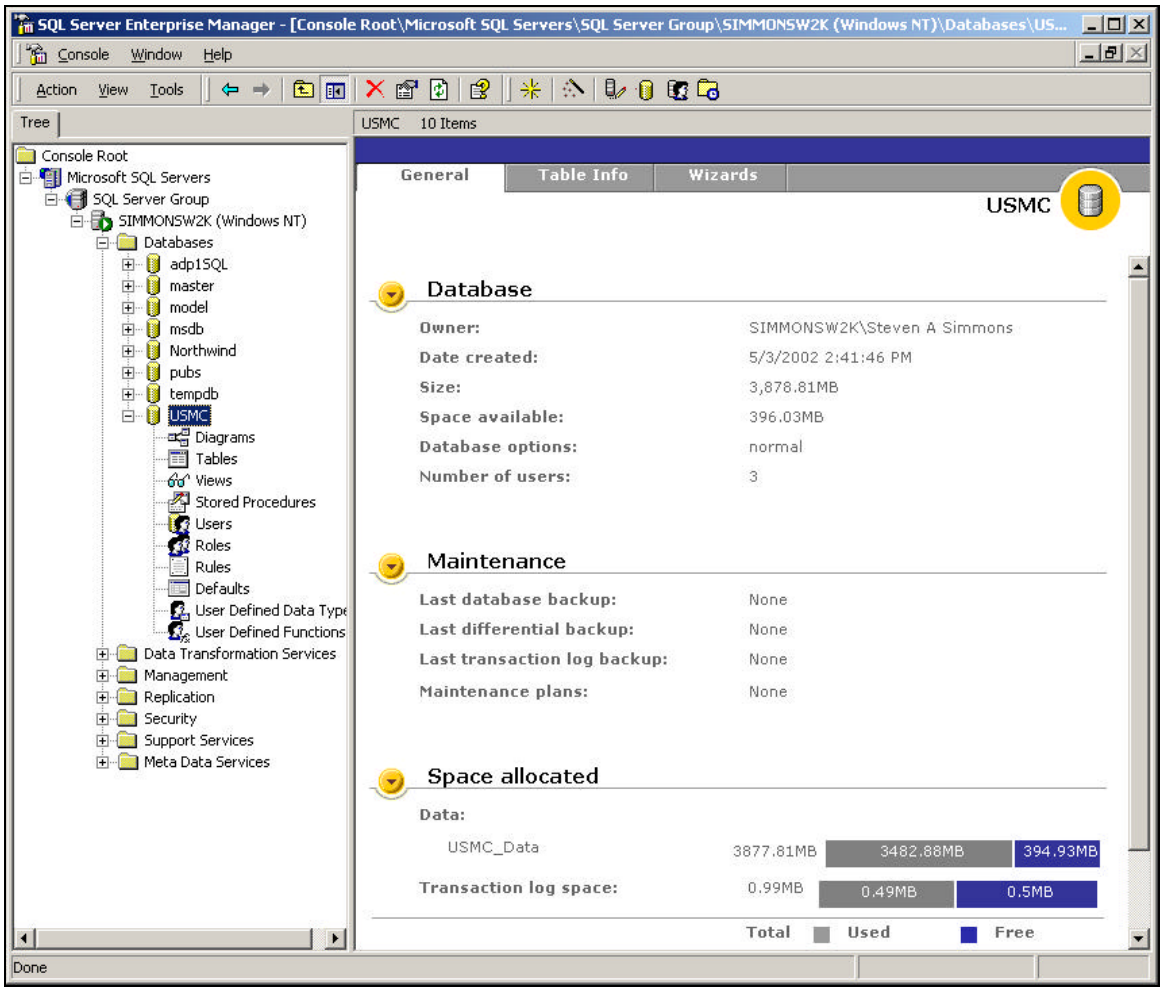


Figure 17. SQL Server 2000 Enterprise Manager Console - TFAS Database

- Maximum database size: 1,048, 516 TB.
- Maximum databases per SQL Server instance: 32,767.
- Maximum number of objects (tables, queries, stored procedures, etc.) per database: 21,474,836,474.
- Maximum number of tables per database: limited by number of objects in database.
- Maximum data fields (columns) per table: 1024.
- Maximum records (rows) per table: only limited by available storage.
- Maximum tables joined in a query: 256.
- Maximum columns per SELECT statement: 4,096.
- Connections. Only limited by number of "software" licenses for connections configured in the operating system.
- Triggers. A Trigger is an event in the database (say a data field update) "triggers" other events (updating other data fields)

- Replication Services. An easy to use wizard guides the database administrator through the setup of keeping two or more databases synchronized. A database in the SQL Server instance may be designated as the "publisher" of data to subscribing databases or as the "subscriber" to some other database. The databases involved in the replication process need not all be SQL Server 2000 databases. Replication can either be "Transactional" (one database is always the source of data to be published) or "Merged" (the data of two databases are combined based on predetermined rules like date time stamps on the data). This feature of SQL Server 2000 will allow the local ODSE to remain synchronized with the national ODSE.
- Windows user account permissions integration. In the figure below, we see that access permissions to our TFAS database named USMC are using Integrated Windows Authentication. In the figure, the administrator is adding a Windows user or group as a SQL Server 2000 user with access permissions to the whole database or to individual tables. Additionally, different access permissions are available for SELECT, INSERT, UPDATE, DELETE, and EXECUTE database transactions.

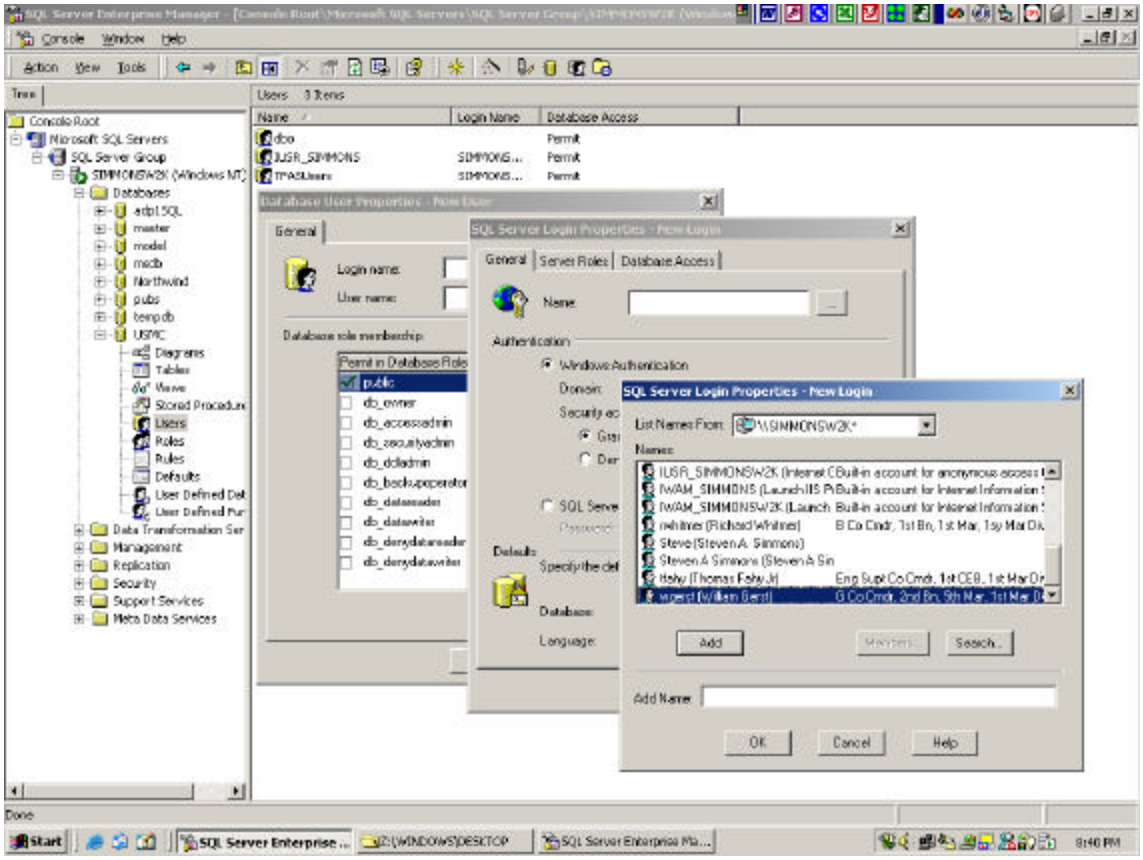


Figure 18. Integrated Windows User Accounts in SQL Server 2000 –TFAS Database

- Database Roles. In addition to standard database task access permissions, different roles can be established. Roles are very similar to Groups in Windows permissions in that the DB administrator can define what actions a group of users can take in the database.
- Stored Procedures (up to 1024 parameters in each stored procedure). Stored Procedures are predefined queries whose values in the WHERE clause are variables that are not defined until run time. Stored procedures can be nested up to 32 levels deep. In the Figure below, we see an example of the GetAlphaRoster stored procedure used in the TFAS web site. This procedure receives values for the Reporting Unit Code (RUC) and Company Code from the web server, performs the query based on these values and returns the record set of data to the web server.

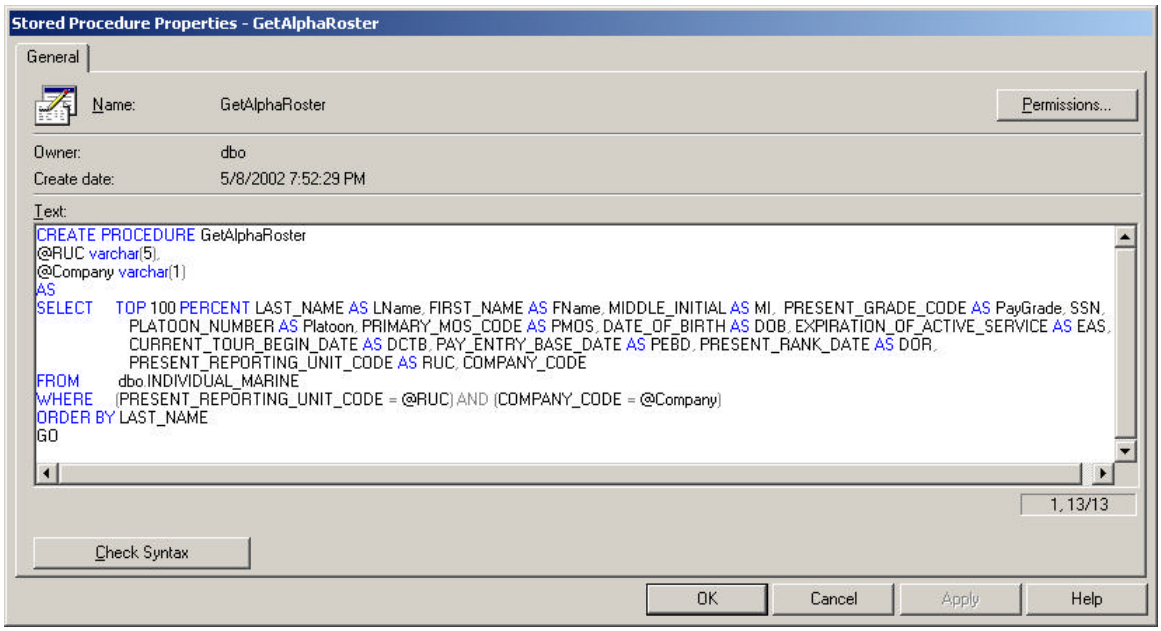


Figure 19. Use of Stored Procedure - TFAS Database

- Database Diagrams. Easy to use interface for viewing the structure of the database and creating relationships among tables. Relationships can be created by dragging and dropping primary keys from one table to the foreign key reference in another table. For complex databases with hundreds of tables, multiple diagrams with differing configurations can be created.
- Support for a wide variety of data types.
- Multiple ways to construct queries: Query Builder Wizards, Query Design Grid similar to Access, and an "English Query" engine for defining queries through English phrases rather than SQL syntax.
- Integrated Support for Visual Basic programming. Many predefined functions are available for common programming tasks (string manipulation, math, etc.). Provides ability to create user-defined functions, which can then be used in stored procedures and triggers.

- Native XML Support. Recordset data returned to the client can be in the form of well-formed XML rather than a traditional recordset.
- Logging. Easy to use interface for viewing the data transactions performed against the database.
- Database Backup Services.
- OLAP Services and Data Warehousing Capbility.

Some of the features described above are not used in our prototype, but could be implemented in further work on the TFAS project or in the actual deployment of TFAS.

In order to access the TFAS data in our development local ODSE, we designated it as a System DSN (Data Source Name) so that it can be accessed by any ODBC compliant application. ODBC software makes it possible to access any data from any application, regardless of which database management system (DBMS) is handling the data. ODBC manages this by inserting a middle layer, called a database driver, between an application and the DBMS. The purpose of this layer is to translate the application's data queries into commands that the DBMS understands. For this to work, both the application and the DBMS must be ODBC-compliant -- that is, the application must be capable of issuing ODBC commands and the DBMS must be capable of responding to them. The application in our case is the ASP web page scripts running on the IIS-5 web server. In the figure below, the registration of the SQL Server 2000 TFAS database as a System DSN for our prototype machine is depicted. Our database is called "USMC" in SQL Server and we have given it the same name (DSN alias) as an ODBC data source.

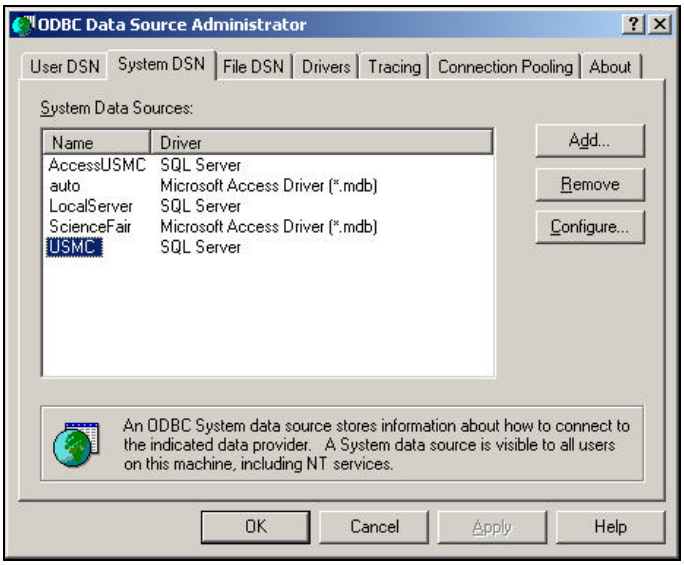


Figure 20. ODBC Data Source Registration – TFAS Database

3. Database Modifications

a. Additional Fields

In order to implement the functional requirements described in Chapter 4, several modifications to the existing ODSE database were required. All of these modifications are additions to the database. No existing fields, tables, or relationships were altered or deleted. The following data fields need to be added to the "Individual_Marine" table:

- Weight
- Height
- BodyFat
- TrouserSize
- BlouseSize
- BootSize
- CoverSize
- Additional Plt Codes and Company Codes

These data fields are commonly tracked by the small unit leader and are defined as required fields in the reports section of our functional requirements. Presently these data fields are not resident in the MCTFS CMF and, therefore, are also absent from the national ODSE. These fields should be added to MCTFS. Multiple pairs of platoon and company codes need to be added to the "Individual_Marine" table. Presently, this table contains the following fields:

- Present_Reporting_Unit_Code
- Temporary_ Reporting_Unit_Code
- Addl_Temp_ Reporting_Unit_Code
- Command_ Reporting_Unit_Code
- FAP_ Reporting_Unit_Code
- Reserve_ Reporting_Unit_Code

A Reporting Unit Code or RUC is the 5-digit numerical code that uniquely identifies a battalion, squadron, or other MCTFS data "reporting" unit. It is the primary identifier for associating a Marine with his/her assigned unit. There are six fields for RUCs for each Marine because at any instant in time the Marine can be assigned to one or many units, each with their own RUC. For instance, a Marine could be assigned permanently to a unit with a RUC of 11111 for a three-year tour of duty, but could also be attending a

school for three months where the school has a RUC of 12345. In this case, the Marine's data record in the Individual_Marine table would have Present_Reporting_Unit Code = "11111" and Temporary_ Reporting_Unit_Code = "12345" while he/she is at the school. The problem with the current schema in MCTFS concerns the existence of only single data fields for Company_Code and Platoon_Code. The data paradox occurs when a Marine is assigned to multiple units during a given time, as in our example. While this Marine is attending the school, what values for company and platoon are resident in MCTFS: the company and platoon for the parent unit, or the company and platoon for the school? The current schema forces the two units in question to repeatedly modify the single company code and single platoon code in MCTFS, as each unit attempts to ensure that their code is resident in MCTFS. We must add Company_Code and Platoon_Code fields for each RUC in MCTFS in order to solve this data dilemma. This issue is critical to the implementation of TFAS at the Echelon II level since queries for data are based on company codes. Adding corresponding company and platoon codes for each RUC in MCTFS will enable the TFAS designer to create platoon data views.

MCTFS programmers at TSO had already anticipated the potentiality of adding additional fields to the MCTFS Central Master File in order to support TFAS functionality. In fact, the need to add additional platoon and company codes in order to clear up the assigned unit conflict in current manpower systems has been known for some time. These updates have not been made due to a lack of funding to pay for them. The funding associated with TFAS implementation should allow for making these needed changes to the MCTFS schema.

b. Additonal Tables

In order to implement seamless TFAS user authentication in Windows, IIS-5, and SQL Server 2000 and by basing all data views on the identity (account username) and unit membership of that TFAS user, we need some mapping between network account user names and the Marines' social security numbers (SSN). This mapping is needed because we must be able to locate the TFAS user as a Marine in the database through the Marine's SSN in the Individual_Marine table. Once the TFAS user is located, we can then read out the user's RUC (code for battalion) and Company_Code. This data can then be stored persistently on the server and used for data queries for the

duration of the user's session. Due to this design requirement, we have added a table to the ODSE database called "TFASUsers." This table contains 3 data fields:

- SSN
- UserName
- CompanyDecsription

The SSN filed is the primary key and has a foreign key reference to the SSN field in the Individual_Marine table. The UserName field is text that contains the network user name for the TFAS user. It only contains the portion of the network user name preceding the "@" symbol. For example, a value for UserName would be "sasimmon" for the network account: sasimmon@nps.navy.mil. The CompanyDescription field is added here in order to have a full English description of the small unit leader's company. By doing this, we are not stuck with the abbreviated company code. Along with adding the TFAS user's network account to the TFAS User's group in Windows, we must also enter the user's data in the "TFASUsers" table in the database in order to allow seamless user access with one network account.

Having defined the distributed architecture of the TFAS prototype as well as its major components at the local level (Network Environment, Web Server, and Backend database), we can now proceed to programming the functionality described in Chapter 4. In the next chapter, a description of the web site layout will be given, as well as several sample portions of ASP code used to build the web pages. Additionally, we will describe in detail how we solved one of our biggest research challenges, how to elicit data from a user in a web environment, translate the data into the MCTFS proprietary unit diary data format, and transmit that data to the MCTFS main-frame for processing.

VI. WEB SITE DESIGN AND PROGRAMMING

In the previous chapters we have defined the functional requirements and system architecture for the TFAS, Echelon II web site prototype. Now that we know the small unit leader's data transaction needs and the detailed systems environment in which the web site will exist, programming the web pages can be accomplished. The purpose of this chapter is to detail the programming efforts made to construct the TFAS, Echelon II web site prototype. This will include a description of our approach to web page authoring, a web site layout, explanation of the login sequence, sample code for a report and data entry task, and the definition of the XML unit diary.

A. WEB AUTHORING METHOD

In programming any complex web site, the programmer(s) must decide what web authoring tools will be used. For large projects, there may be multiple programmers with a variety of needs and skills. The functions to be implemented may require lengthy sections of code. The choice of the right tool can significantly reduce time spent on complex yet common and repetitive tasks. Additionally, version control must be strictly enforced during the group's collaboration. For these reasons, Microsoft's Visual InterDev 6.0 IDE (Integrated Development Environment) was selected as the web-authoring tool for the prototype's development. InterDev has many features to support the individual programmer as well as group projects. These features are as follows:

- WYSIWYG Editor. Provides a "What You See Is What You Get" interface where the web author can type text as in a word processor and drag and drop complex objects for dynamic content.
- Design, Source Code, and Quick View modes.
- Site Designer. Web site building using drag and drop graphic icons and diagrams.
- Server and Client side scripting libraries based on the W3C Document Object Model (DOM) with code-in-site.
- Team development with local and master mode, and version control.
- Data bound Design Time Controls (DTCs) for rapid web-enabled database development.
- Built-in extensive library of web site themes.
- Cascading Style Sheet authoring.

- XML integration through the MSXML 4.0 API. MSXML is Microsoft's API (Application Programming Interface) for the implementation of the W3C's XML 1.0 specification.
- Support for COM object development and integration in web applications.
- Debugging tools.
- Integrated ActiveX Controls library.
- Remote authoring and deployment capability.
- Integration with other Microsoft products, namely the Windows NT/2000 operating systems, IIS-5 web server, and SQL Server 2000 database.

Figure 21 below shows an example of the Visual InterDev Integrated Development Environment (IDE). In the figure, the AlphaRoster.asp web page is depicted in Source Code view. The data to populate this report is generated through design-time controls (drag and drop objects), which are data bound to a specified stored procedure. A more detailed description of the code will be presented later in this chapter.

The choice of a particular web-authoring tool does lock the web developer into a particular technology. In developing the prototype, the selection of InterDev 6.0 locks us into the ASP scripting technology and a dependence on InterDev server extensions. Such a selection could cause the prototype web site to become obsolete as newer technologies evolve. Just this past year, Microsoft introduced its .NET Framework for integrated software development. The .NET Visual Studio IDE, released to support this initiative, contains a web development application. This application, however, is essentially a new, improved InterDev 6.0 and is still based on ASP technology. The .NET IDE interface is virtually identical to InterDev 6.0, but many XML-based design tools have been added. The tools and techniques (Design Time Controls) used in the development of the prototype are still available and are supported by .NET Visual Studio. Any application developed in InterDev 6.0 will still be functional in .NET studio. Therefore, the system architecture and programming code will not be affected by the adoption of .NET Visual Studio as a web-authoring tool.

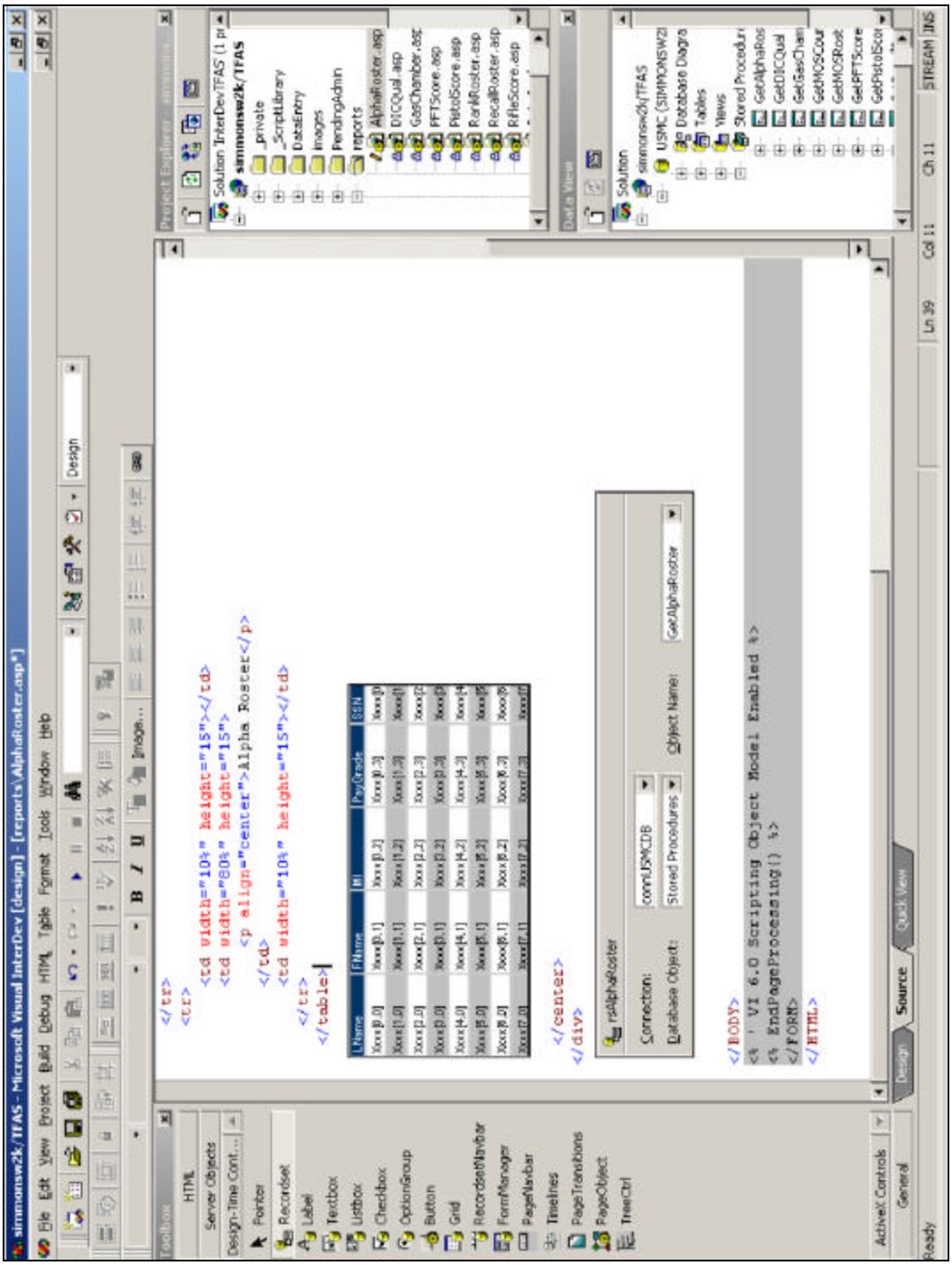


Figure 21. Visual InterDev 6.0 Web Authoring Tool

B. WEB SITE LAYOUT

With any complex web site, it is good practice to plan the directory structure before actually coding the web pages. Most web site designers create a storyboard which

depicts each web page, how they interact, and the overall hierarchical structure of the site. In Figure 22 below, a site layout diagram of the Echelon II web site prototype is depicted. The structure of the web site is based on the user functional requirements defined in Chapter 4. The top-level directory named "TFAS" is contained within root directory of IIS-5, the "WWWROOT" directory. The subdirectories within the TFAS directory mirror the major groupings of functional requirements for the various types of TFAS users at the small unit level. By structuring the web site in such a manner, it will be easy to assign varying access permissions for TFAS users with different authority to view data or conduct data entry.

The default web page for the TFAS web site or home page is file "index.htm." This web page defines the structure for the viewing area in the client's browser by dividing screen real estate into four frames. These four frames are constructed initially from four files: Banner.htm, Footer.htm, WelcomeDisplay.asp, and Menu.asp. The banner and footer are static HTML pages and contain links to Marine Corps web pages and a standard USMC disclaimer. The Menu frame contains a dynamic expandable tree menu whose nodes contain the links to all of the reports and data entry pages. The frame containing the output from the WelcomeDisplay.asp file takes up 80% of the screen and is used as the target frame for all web content during the session. Initially, this frame displays a TFAS logo and a welcome message tailored to the individual user. This welcome page could be augmented in the future to notify the user if he/she has any pending administrative actions to perform, like approving a leave request. A detailed description of the login sequence is given later in this chapter, which further explains the interaction of this framed user interface. Screen shots of the prototype will also be shown.

It should be noted that the design of the user interface and its esthetics were modeled after the guidelines set forth in References 10 and 11, the Marine Corps' guidelines for publicly accessible web sites. The web pages were designed with minimal graphics in order to speed the downloading of files.

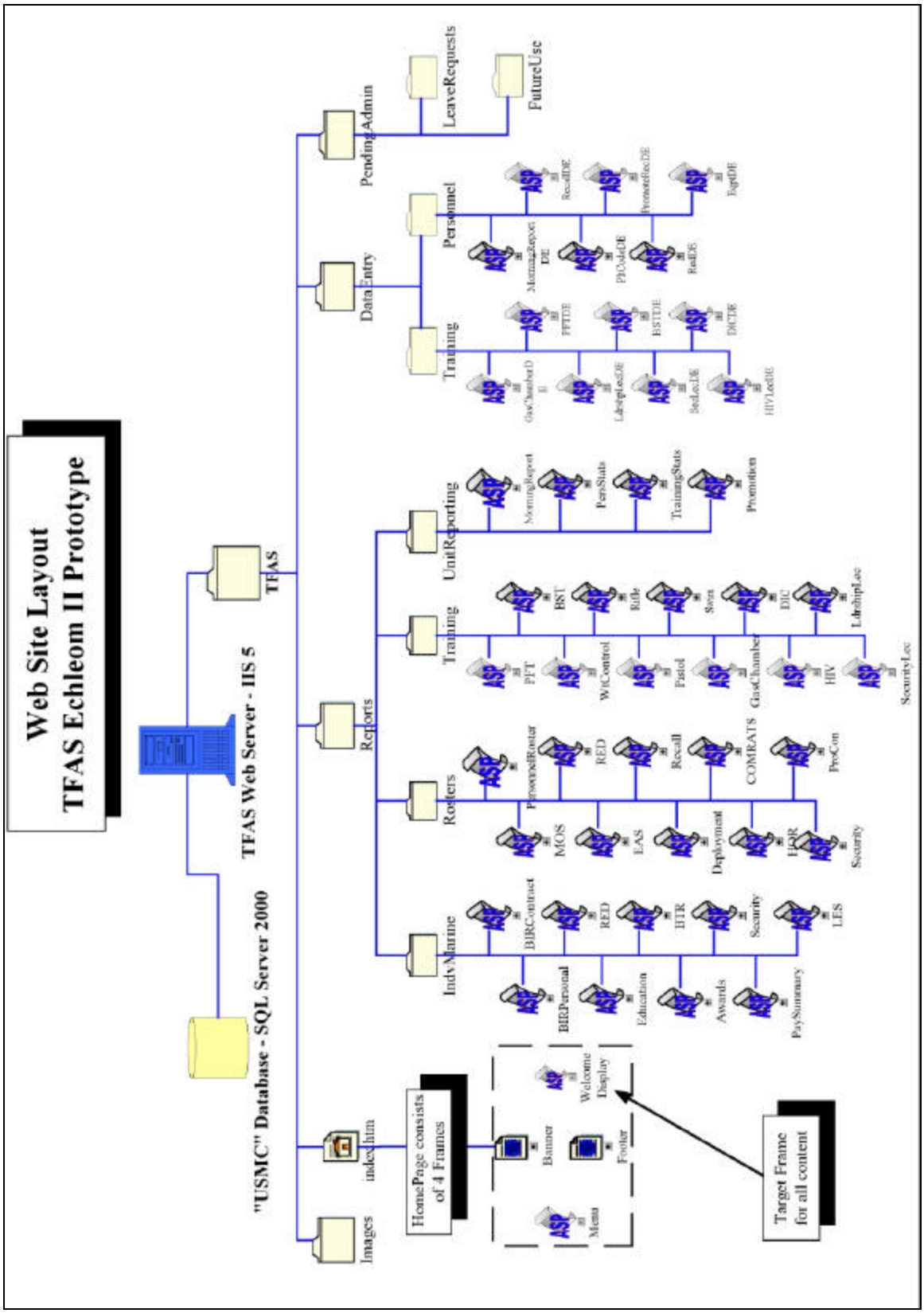


Figure 22. Web Site Layout – TFAS Echelon II Prototype

C. LOGON SEQUENCE

1. General Description

The key to implementing the functional requirement of creating web access to manpower data for the small unit leader and restricting the data to only Marines in the leader's unit is the ability to bind the identity of the leader dynamically at run time to queries for the data. Additionally, the web site must deny access to all unauthorized users and restrict access to certain TFAS users with limited permissions for certain TFAS data tasks. Accordingly, the TFAS web site prototype does not allow anonymous access. To implement these requirements, there must be some type of logon sequence when the user first enters the TFAS web site. Only users with network user account and access permissions on the files of the web site and on the SQL Sever 2000 will be allowed access. By using Windows user accounts for permissions to all of the components of the TFAS system, the user only needs one username and password, their network account.

There are two options for using Windows user accounts for permissions in IIS-5, Basic Authentication and Integrated Windows Authentication. [Ref 13] Basic Authentication sends the username and password in the clear over the network, an obvious security risk. In Integrated Windows Authentication, the web browser encrypts the user name and password before transmission to the server. While Integrated Windows authentication seems the obvious choice for security reasons, the prototype implements Basic Authentication. The reason for using Basic Authentication is that only the Internet Explorer web browser supports Integrated Windows Authentication. The use of Basic Authentication allows for TFAS access by all web browsers. In addition, SQL Server access permissions presently work only with Basic Authentication for web connections to the database. By using Basic Authentication, the prototype is now vulnerable to packet sniffers and other security threats. However, these threats can be alleviated by encrypting all transactions via Secure Socket Layer, thus protecting user accounts and passwords during transmission between the client machine and the server. It should be noted that by using Basic Authentication, the client is always prompted for their network username and password, whereas in Integrated Authentication, the browser uses the network username and password kept in memory from when the user logged onto the workstation.

2. Sample Logon

The figures below show the logon sequence implemented on the Echelon II web site prototype. Following the figures, a portion of the code is shown whose function is to "grab" the user's information and store it persistently on the server so it can be used for queries for data.

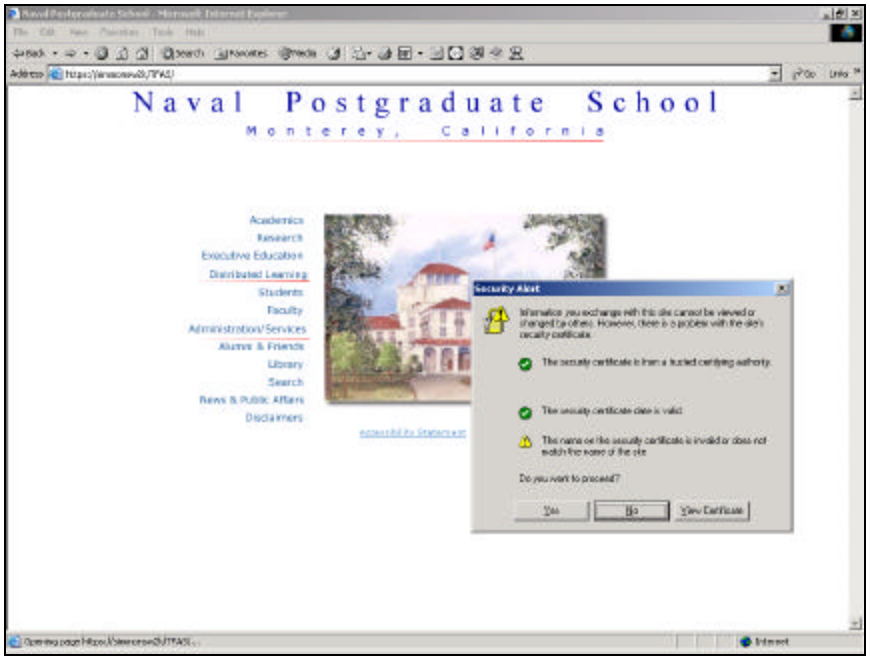


Figure 23. TFAS Logon – Enable Encryption

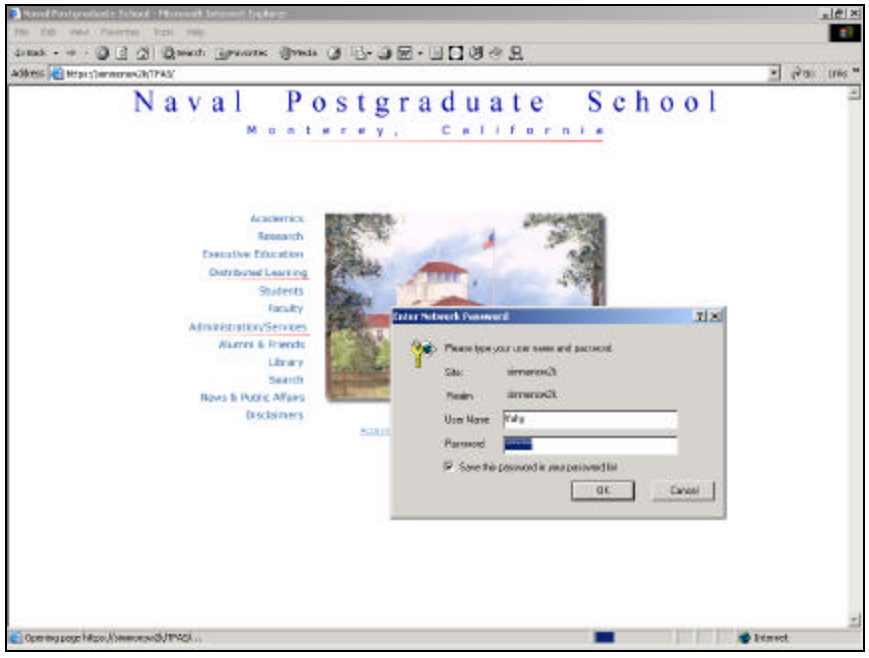


Figure 24. TFAS Logon – Enter Network Password

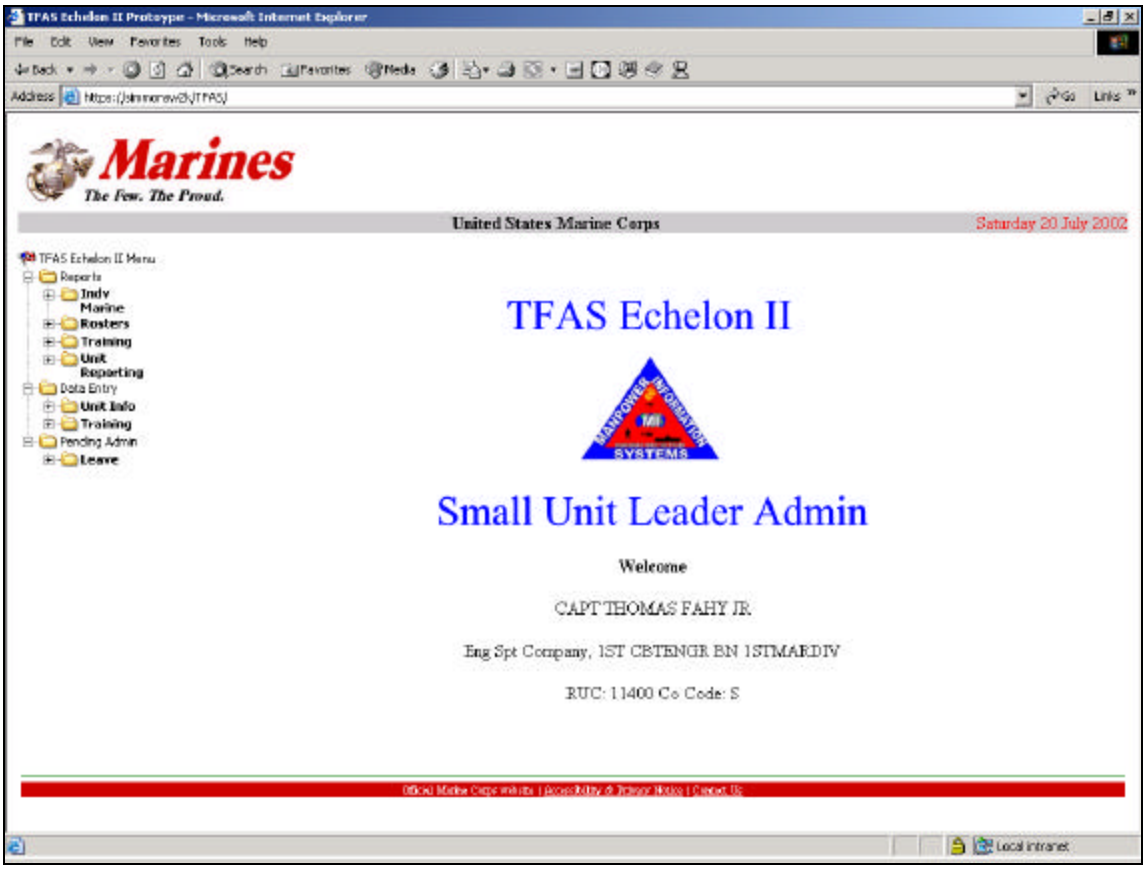
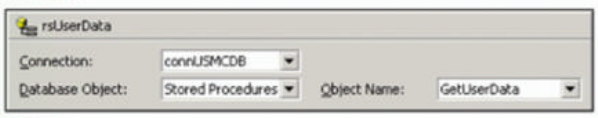


Figure 25. TFAS Logon – Web Site Welcome Screen

Once the TFAS user enters their network username and password, they are encrypted and transmitted to the server. At the server, they are decrypted and then IIS-5 checks all Windows and IIS-5 permissions on requested files for the user. If that user does not have access permission to files, they are denied access to the web site. If they do have access permissions, the web server returns the HTML files to the client and runs any of the ASP pages. The key page in this logon sequence is the WelcomeDisplay.asp page. When the server loads this page and begins to run its VBScript, the server encounters a section of code that must be executed before returning any output to the client. In this code, the username is retrieved from the Request.ServerVariables object. Then, a recordset object is opened. This recordset is based on a stored procedure in the SQL Server 2000 database that queries data from the "TFASUsers" and "Individual_Marine" tables. The variable passed to the stored procedure is the username. Since all network usernames are unique within a domain, there will only be one tuple of data returned.

```
<%@ Language=VBScript %>
<SCRIPT id=DebugDirectives runat=server language=javascript>
// Set these to true to enable debugging or tracing
@set @debug=false
@set @trace=false
</SCRIPT>
<% ' VI 6.0 Scripting Object Model Enabled %>
<!--#include file="_ScriptLibrary/pm.asp"-->
<% if StartPageProcessing() Then Response.End() %>
<FORM name=thisForm METHOD=post>
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>
<SCRIPT LANGUAGE=vbscript RUNAT=Server>

        'get the Win NT/2000 UserName from the requesting client
        dim userName
        userName = Request.ServerVariables ("LOGON_USER")
        dim index
        index = InStr(userName, "\")
        userName = Mid(userName, index + 1)

        'Query database for User's data & set session variables
        rsUserData.open
        rsUserData.moveFirst

        'Set Session Variable values
        Session("userName") = userName
        Session("fname") = trim(cstr (rsUserData.fields.getValue("FIRST_NAME")))
        Session("lname") = trim(cstr (rsUserData.fields.getValue("LAST_NAME")))
        Session("MI") = trim(cstr (rsUserData.fields.getValue("MIDDLE_INITIAL")))
        Session("FI") = trim(cstr (rsUserData.fields.getValue("FIRST_INITIAL")))
        Session("SSN") = trim(cstr (rsUserData.fields.getValue("SSN")))
        Session("rank") = trim(cstr (rsUserData.fields.getValue("GRADE_ENGLISH_DESCRIPTION")))
        Session("coCode") = trim(cstr (rsUserData.fields.getValue("COMPANY_CODE")))
        Session("RUC") = trim(cstr (rsUserData.fields.getValue("PRESENT_REPORTING_UNIT_CODE")))
        Session("battalion") = trim(cstr (rsUserData.fields.getValue("UNIT_NAME")))
        Session("company") = trim(cstr (rsUserData.fields.getValue("CompanyDescription")))

        'Build String for use in HTML Output
        dim name
        dim unit
        dim unitCode

        name = Session("rank") & " " & Session("fname") & " " & Session("lname")
        unit = Session("company") & ",  " & Session("battalion")
        unitCode = "RUC:  " & Session("RUC") & "  " &  "Co Code:  " & Session("coCode")
</SCRIPT>

<p align="center"> </p>
<p align="center"><IMG height=217 src="images/TFASEchII.gif" width=389 border=0></p>
<P align=center><STRONG>Welcome </STRONG> </P>

<SCRIPT LANGUAGE=vbscript RUNAT="Server">
    'HTML Ouput to display a customized Greeting
    Response.Write "<P align=center>"
    Response.Write name
    Response.Write "</P>"

    Response.Write "<P align=center>"
    Response.Write unit
    Response.Write "</P>"

    Response.Write "<P align=center>"
    Response.Write unitCode
    Response.Write "</P>"

</SCRIPT>
```




```
</BODY>
<% ' VI 6.0 Scripting Object Model Enabled %>
<% EndPageProcessing() %>
</FORM>
</HTML>
```

Figure 26. Programming Code – User Logon

Each column of data in the recordset is programmatically read out and stored in its own session variable. This data includes the TFAS user's First Name, Last Name, MI, FI, Rank, SSN, RUC, Company Code, and English versions of the Company and RUC (battalion) codes. This data is kept in session variables stored in memory on the server where it can be used during the session to tailor the web site to this user. This data is also used when constructing the web unit diaries for data entry.

D. DATABASE CONNECTIVITY AND WEB INTEGRATION

By selecting a web authoring tool like InterDev 6.0 to build the Echelon II web site prototype, many of the repetitive programming code required for database connectivity and recordset construction can be built automatically by using many of InterDev's wizards and built-in Design Time Controls (DTCs). DTCs are graphic icons that can be dragged and dropped on the source code page. While graphic in appearance to the web author, the DTC icons really represent chunks of ASP code that will execute the desired task.

Unlike other web authoring tools where the web author commonly encodes a database connection on every page, InterDev can create one database connection for the entire web application and reuse that connection on any page needing data from the database. In the figure below, a screenshot of the WelcomeDisplay.asp page in the InterDev web page design tool is shown. In the figure, the various components used for database connectivity for the web application and WelcomeDisplay.asp page can be seen. The actual code for the database connection for web application is kept in a file called "Global.asa." The Global.asa file is a special ASP page that is called by the web server whenever the web site is accessed by the client. In this file, global tasks for the web application are performed, like connecting to a database. Additionally, the Data View window can be used to see a list of all available database objects. The web author can also view live data from any of these objects, as well as construct custom queries not available in the database.

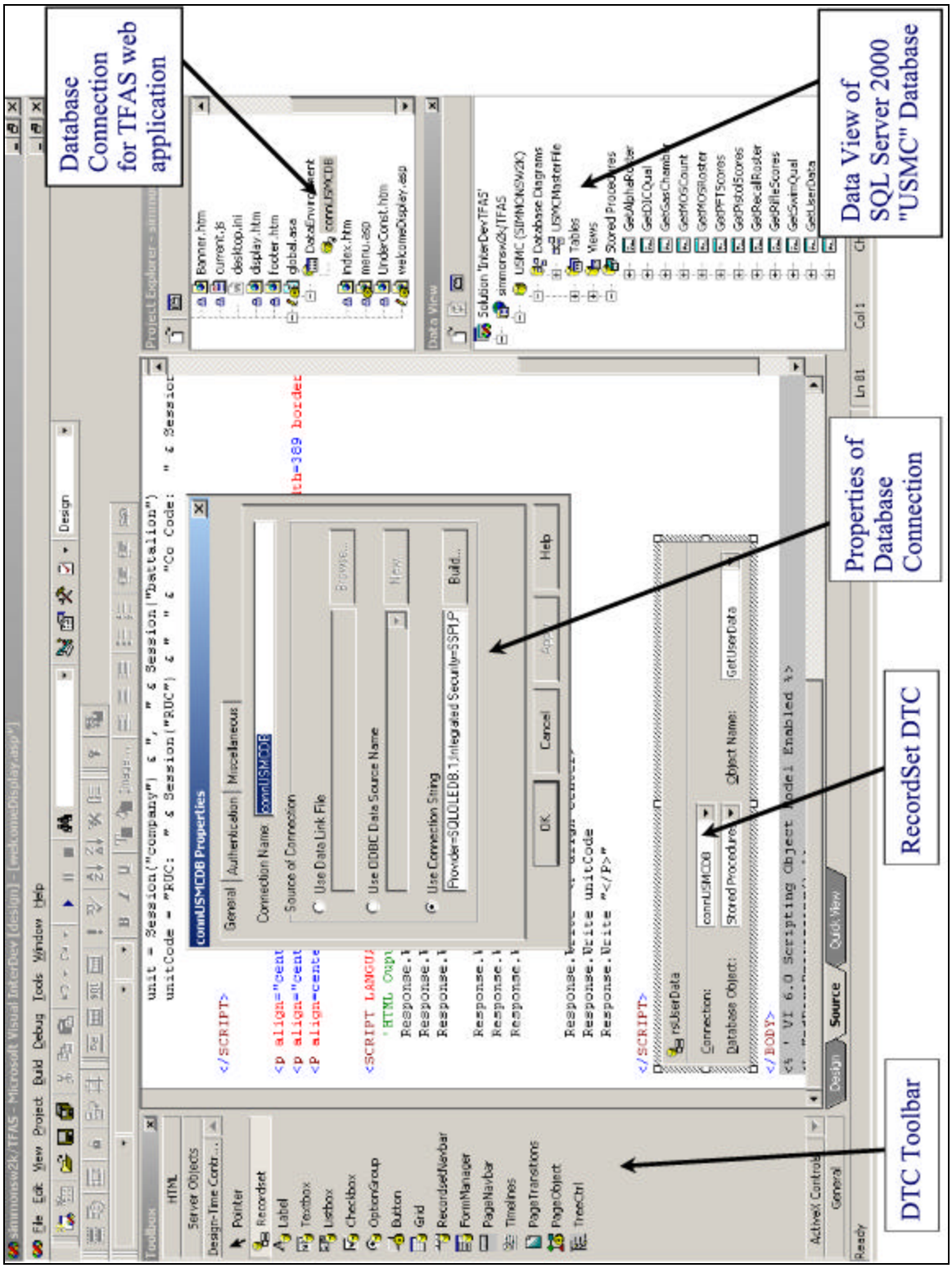


Figure 27. Database Connectivity Design

When examining the code in the figure above, there appears to be no code for creating the recordset "rsUserData" needed in the WelcomeDisplay.asp page. The programming for this recordset is present, however, it is just hidden in the graphic icon named rsUserData near the bottom of the page. Once a database connection is created for the application, the web author drags a recordset DTC onto the source page. To bind the recordset to a particular table, query, or stored procedure, the web author simply right clicks the recordset DTC and selects the Properties menu item. In the resulting Properties dialog box, the web author selects the database connection and object to bind to the recordset DTC. If the recordset uses a stored procedure, parameter values are also set. In Figure 28 below, the property settings are shown for the recordset, rsUserData used on the WelcomeDisplay.asp page.

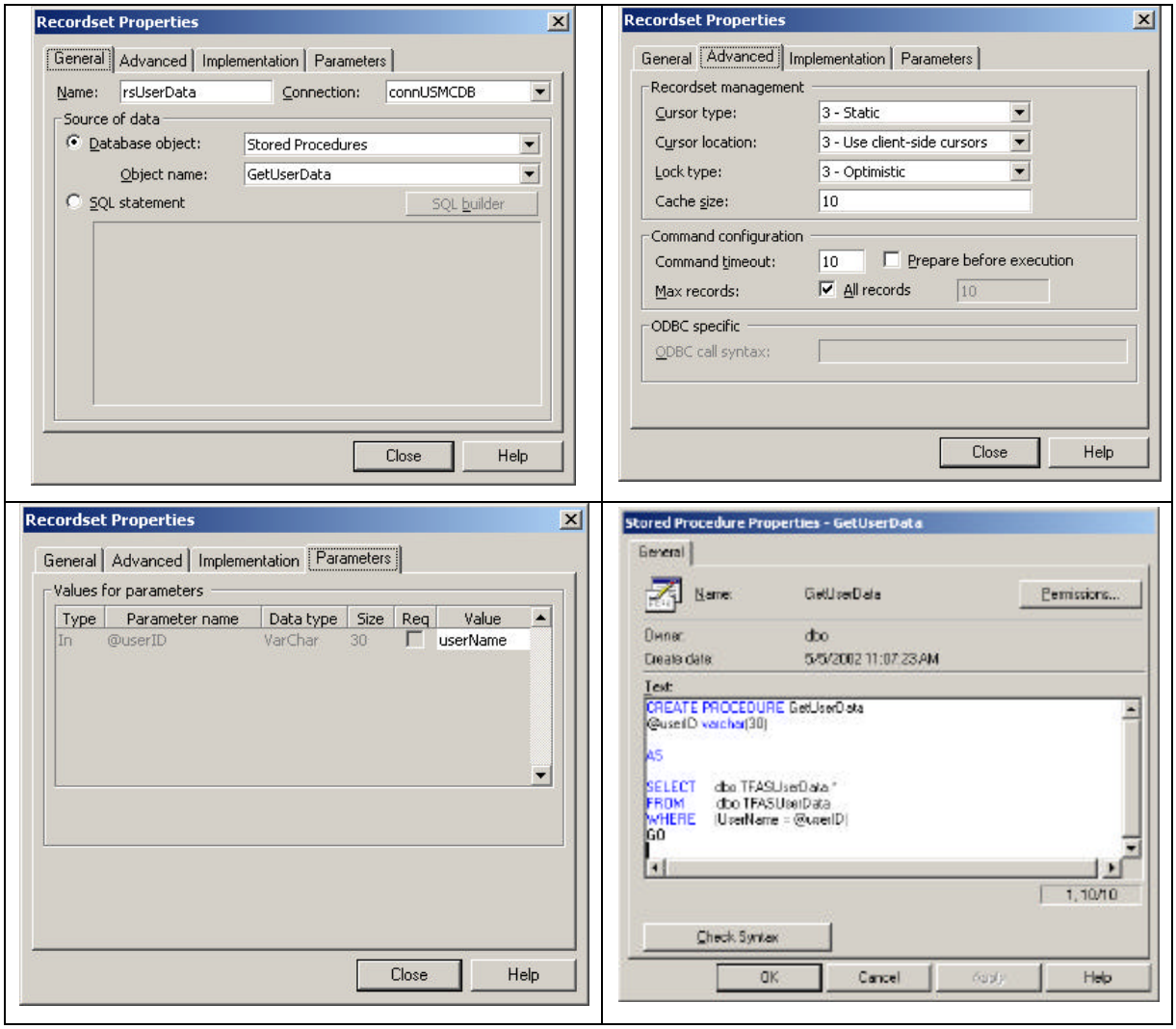


Figure 28. Database Recordset Design

The recordset rsUserData makes use of the database object named "GetUserData." GetUserData is a stored procedure in the SQL Server 2000 database. It has one variable named UserID whose value is defined at run time. The value of the parameter is the string containing the TFAS user's network username. The stored procedure calls a query named "TFASUserData" and returns only those records containing data for the TFAS user. In the figure below, the query TFASUserData is depicted. This query joins several of the ODSE tables and the table called "TFASUsers" in order to obtain the data needed to set values for the session variables.

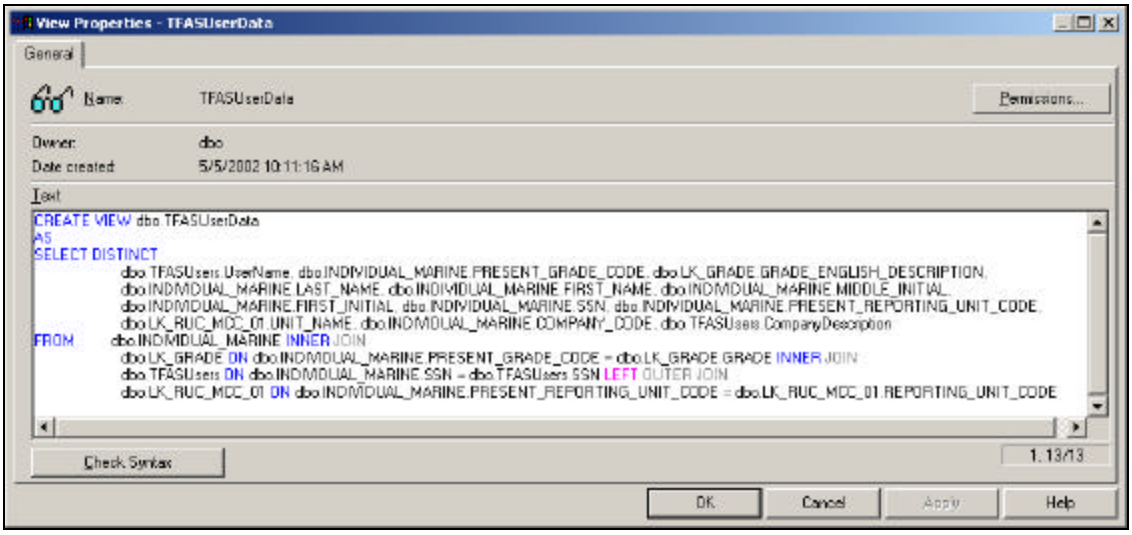


Figure 29. TFASUserData Query – SQL Server 2000

The technique demonstrated above for database connectivity and data retrieval is representative of the technique used in all the web pages in the prototype. In the sections below, a description of a sample report and a data entry task is presented. For these samples, the database connectivity techniques will not be shown, as they are identical to those shown in Figure 29.

It should be noted that in order to use these Design Time Controls available in InterDev 6.0 in the deployed TFAS, Echelon II web site, InterDev 6.0 extensions must be installed on the web server. These extensions are merely "dll" files for the COM objects used by the web server and operating system to execute the programming functionality defined by the DTCs. InterDev 6.0 server extensions are analogous to Front Page server

extensions, which enable many of the user-friendly web authoring tools available in Microsoft Front Page.

E. SAMPLE REPORT

All of the read-only report functional requirements defined in Chapter 4 are designed in the same manner. A separate ASP page creates each report. A link to activate the page is listed in the nodes of the menu available in the left portion of the TFAS web interface. Clicking the desired node will cause the server to run that ASP on the server and deliver the output of the ASP to the target "display" frame. For example, if the TFAS user wants to view a report of the unit's current PFT scores, clicking the node under Reports/Training in the menu will cause the PFT.asp file to be run on the server and a table of PFT data is returned to the display frame. In the figure below, a screenshot of this process is depicted.

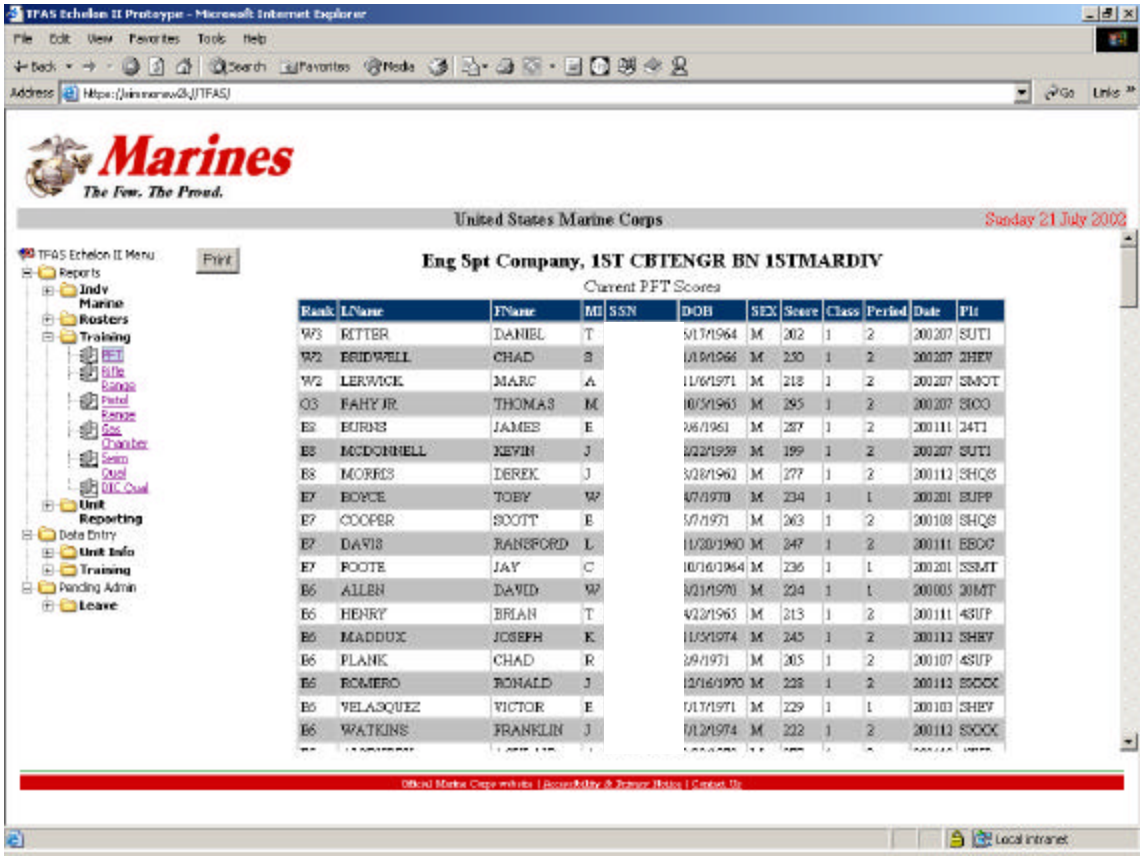


Figure 30. Sample TFAS Report - PFT

The PFT.asp file has a recordset DTC that is bound to a stored procedure in the SQL Server 2000 database called "GetPFTScores" whose parameters are RUC and CoCode. These parameters define which records are to be returned to the server – only those records for the TFAS user's battalion (RUC) and company (CoCode). In this stored procedure, the Individual_Marine and Training ODSE tables are "joined" to get the required data for the PFT Report. The stored procedure GetPFTScores is presented in the Figure 31 below. Notice also that we have filtered out all Navy personnel attached to the unit. This filtering is accomplished by requiring that any records retrieved will not have a SSN that begins with "N". Navy personnel attached to Marine Corps units are not required to take the semi-annual PFT, so their data should not be displayed in the PFT report. In MCTFS, a leading "N" attached to the SSN data field easily identifies Navy personnel.

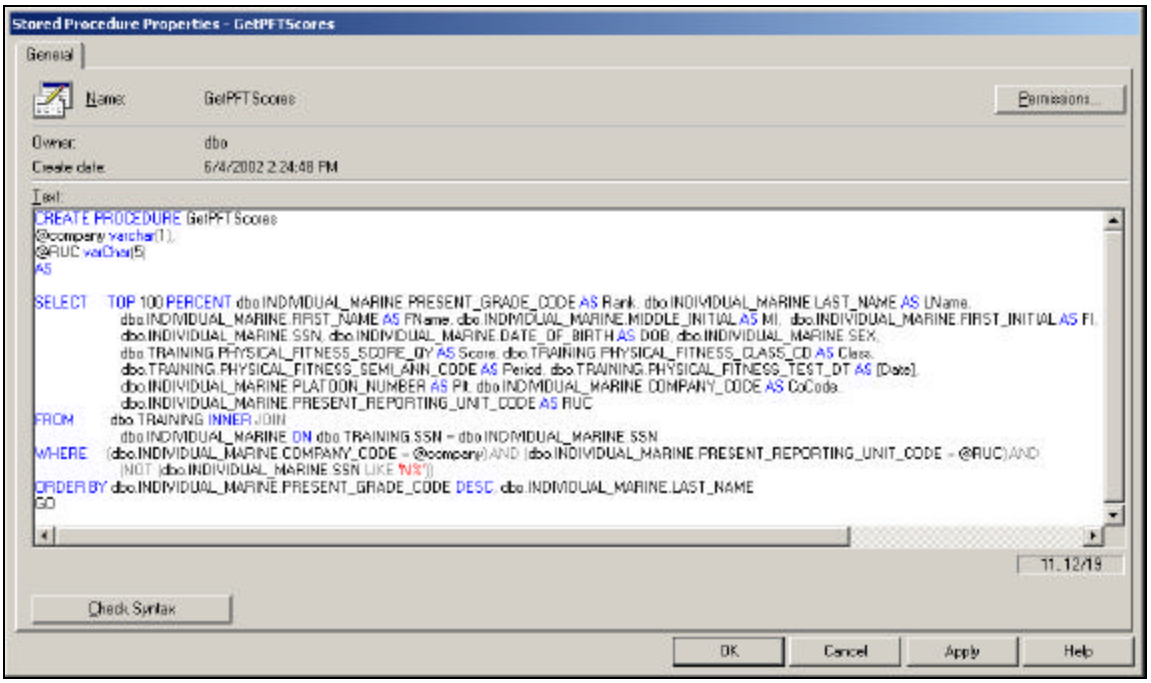


Figure 31. Stored Procedure for PFT Report

F. XML UNIT DIARY

Before a sample of the data entry task is presented, it is important to understand how the TFAS web application takes user input from a web page and translates this data

into a “unit diary.” As presented in earlier chapters, the unit diaries transmitted to the MCTFS mainframe for processing are in a proprietary format understood only by MCTFS systems. Rather than programming a web script to translate user data into this MCTFS unit diary format, the prototype will transform user data into a predefined XML document. When the XML document or “XML Unit Diary” is created on the server, it can be stored in a network directory. Later, these XML diaries can be loaded into the UDMIPS application where they are transformed into the MCTFS unit diary format. Then, these web diaries, along with all of the “normal” diaries created by administrative personnel in UDMIPS, are transmitted in a “batch” to the MCTFS collection server where they will be processed.

XML or Extensible Markup Language is a formal recommendation from the World Wide Web Consortium (W3C). XML is similar to the language of today's Web pages, the Hypertext Markup Language (HTML). Both XML and HTML contain markup symbols to describe the contents of a page or file. The major difference is that HTML describes the content of a web page (mainly text and graphic images) only in terms of how it is to be displayed (for example, the letter "p" placed within markup tags starts a new paragraph in a HTML web page). XML, on the other hand, describes the content in terms of what data is being described. XML is a language for writing a language to organize data and to describe the meaning of the content of the data.

The personnel at TSO in Kansas City, MO have defined a schema for the XML diary and a summary of this schema is presented in the following table. This schema defines the “language” for a unit diary.

Content Model	Occurrence Cardinality	Data Content Required	Data Content	Child Nodes
VersionID	1	Required	Character	None
SystemID	1	Required	Character	None
SystemCode	1	Required	Character	None
Unit	1	Required	Character	None
CalendarYear	0-1	Optional	Character	None
DiaryNumber	0-1	Optional	Character	None
DiaryType	0-1	Optional	Character	None
DiaryDate	1	Required	Character	None
DiaryClass	0-1	Optional	Character	None
Preparer	1	NA	Node	Member
Certifier	1	NA	Nodes	Member

Content Model	Occurrence Cardinality	Data Content Required	Data Content	Child Nodes
				Certifier.Title Certifier.GradeComponent
Notes	1	NA	Nodes	Pereparer.Notes Reviewer.Notes Certifier.Notes
EventRUC	0-1	Optional	Character	None
DiaryTransactions	1	NA	Nodes	DiaryTransaction
Member	1-M	NA	Nodes	SSN LastName Initials
SSN	1	Required	Character	None
LastName	1	Required	Character	None
Initials	1	Required	Character	None
Certifier.Title	0-1	Optional	Character	None
Certifier.GradeComponent	0-1	Optional	Character	None
Preparer.Notes	0-1	Optional	Character	None
Reviewer.Notes	0-1	Optional	Character	None
Certifier.Notes	0-1	Optional	Character	None
DiaryTransaction	1-M	NA	Nodes	TypeTransactionCode TypeTransactionSeq Member Preparer Responses HistoryStatement
TypeTransactionCode	1	Required	Character	None
TypeTransactionSeq	1	Required	Character	None
Responses	1	NA	Nodes	Response
HistoryStatement	0-1	Optional	Character	None
Response	1-M	NA	Nodes	WordType ResponseValue
WordType	0-1	Optional	Character	None
ResponseValue	1	Required	Character	None

Table 11. XML Unit Diary Schema [After: Ref. 17]

G. SAMPLE DATA ENTRY

Using the XML schema defined above as a template, construction of an XML Unit Diary can be accomplished. During a data entry transaction in the Echelon II prototype, the TFAS user will click one of the links in the Menu under the Data Entry menu node. For example, if the user wants to enter PFT scores for members of the unit, he/she will select the PFT link under Data Entry/Training in the menu. Activating this link will cause the server to run the UpdatePFT.asp file on the server. Before returning any content to the target display frame, a recordset of PFT data is retrieved from the database for all members of the units. This data (Rank, Fname, Lname, SSN, etc.) is

read-only and is the data context for the data entry task. The technique for retrieving this data is identical to the one described above for the PFT report. Unlike the report, where all of the unit member's data was displayed to the screen, this page will display the data one record at a time. Navigation buttons are provided which enable the user to page through each individual Marine's data. Standard HTML form components are added to provide a means by which the PFT score data may be entered. Lastly, a Submit button is added to the page to activate transmission of the entered data to the server. In the figure below, a screenshot of the data entry page for PFT scores is shown.

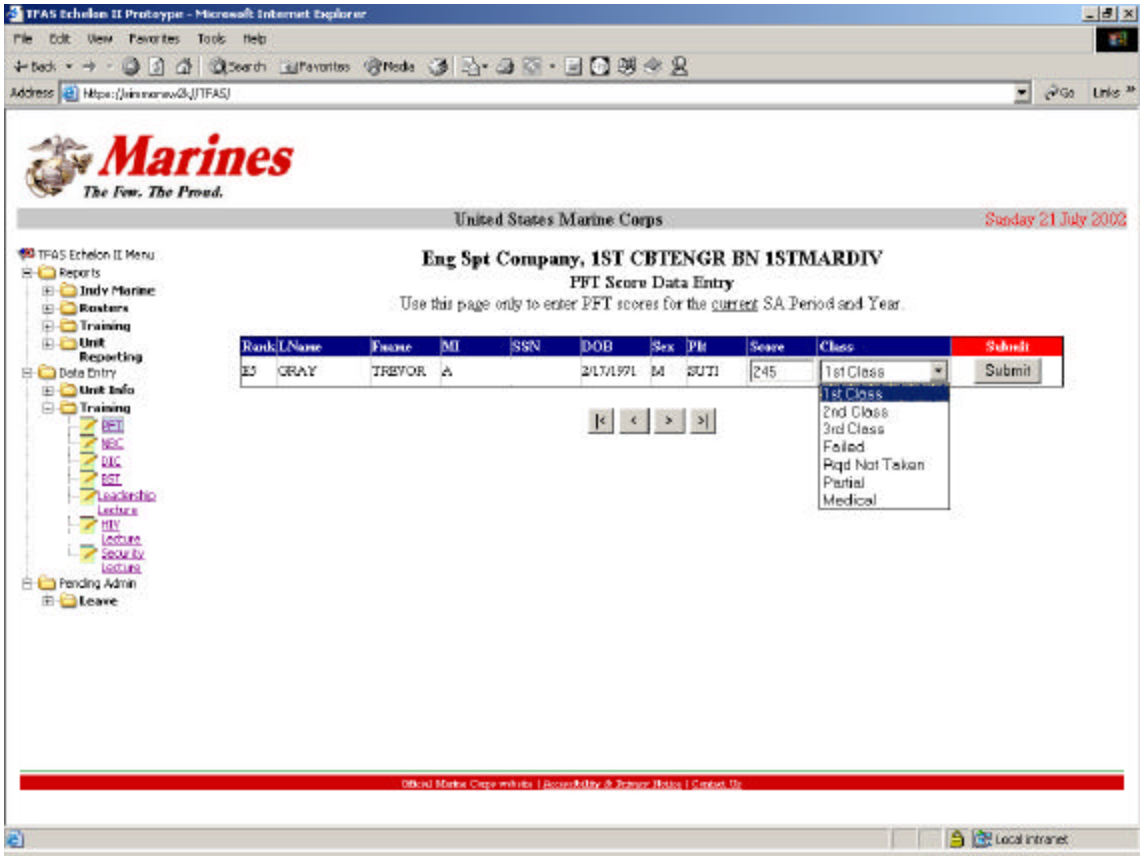


Figure 32. Sample Data Entry – PFT

When the user enters data for a particular Marine and presses the Submit button, the data entered is returned to the server where a VBScript subroutine called "cmdSubmit_onclick" is run. This subroutine performs several tasks. First the values of the data are checked for valid data. PFT Scores can only be numerical data between 0

and 300. If invalid data is detected, an error message is returned to the user and the subroutine finishes. If the data submitted is valid, then the subroutine continues and creates the XML Unit Diary.

To create the XML diary file, the subroutine first gathers all of the required data from the session variables, recordset data for the subject Marine, and other system data. This data is stored in VBScript variables and is used to set values in various character data nodes of the XML Unit Diary. Once the data is prepared, the subroutine then creates an instance of an XML Document Object Model (DOM) available in the MSXML 4.0 API. After creating the XML DOM object, the subroutine then creates each node in the XML Unit Diary schema and sets their values with data. Once the entire XML diary object is populated with data, the object is written out to an XML file in a specified network directory. The name of the file is generated dynamically and consists of the diary date, diary type, and SSN of the subject Marine. By using this naming convention, it will be easy to search the current diary directory or a collection of past diaries for a specific web diary transaction. This will facilitate troubleshooting of failed diaries. Figure 33 shows the network directory where the web generated XML diaries are stored.

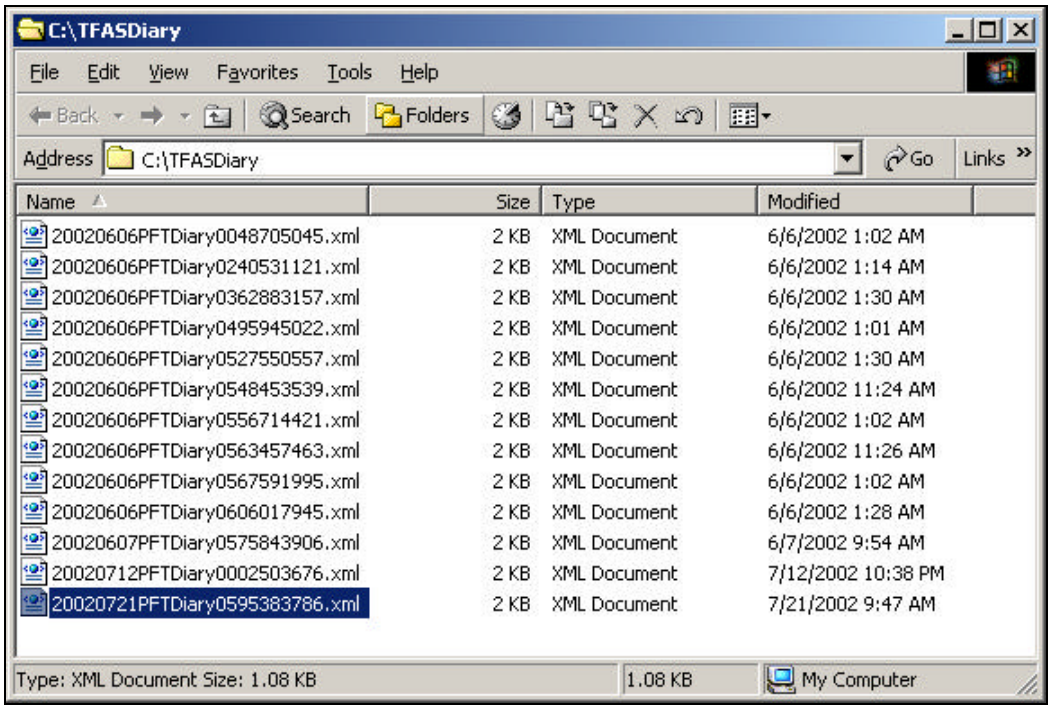


Figure 33. XML Unit Diaries in Network Directory

In the example directory above, several XML Unit Diary files are depicted. It can be readily determined that these files were created on different dates. In the deployed model of the TFAS, Echelon II web site, this directory would be cleared at the end of each business day as the files are imported into UDMIPS for translation. The files would then be moved to another directory in order to keep a historical record and could be used for trouble shooting failed diary transactions.

After the XML diary file is written out to the network directory, a notification is sent back to the user's browser that the diary was successfully created. If there were any errors during the diary creation process, then the user is notified. Figure 34 shows an example of user notification for the successful creation of a unit diary.

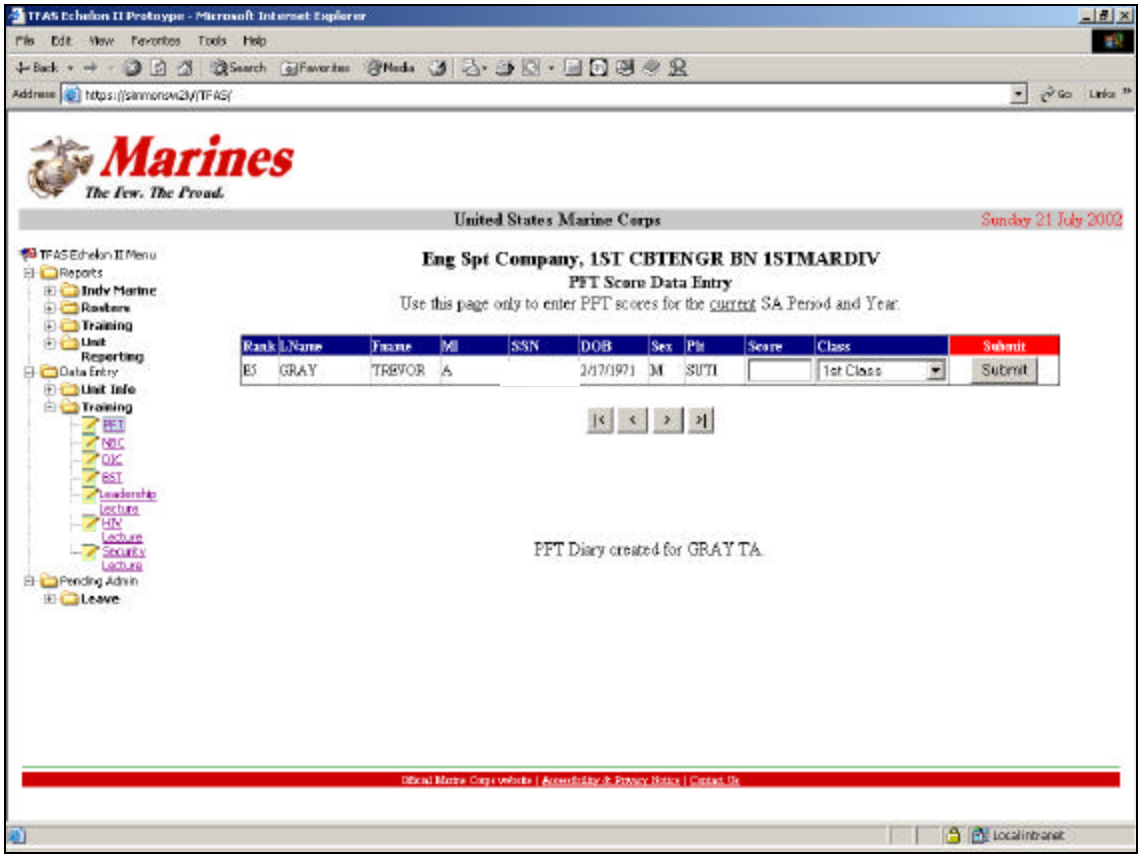


Figure 34. User Notification of Successful Diary Submission - PFT

The XML file generated by the data entry task is populated with all of the required data for the data update to be processed on the MCTFS mainframe. Figure 35 shows a sample XML diary file for the PFT data entry transaction as viewed in a browser. Notice the values of the data in the nodes correspond to the data for the subject Marine as well as data relative to the TFAS user entering the data.

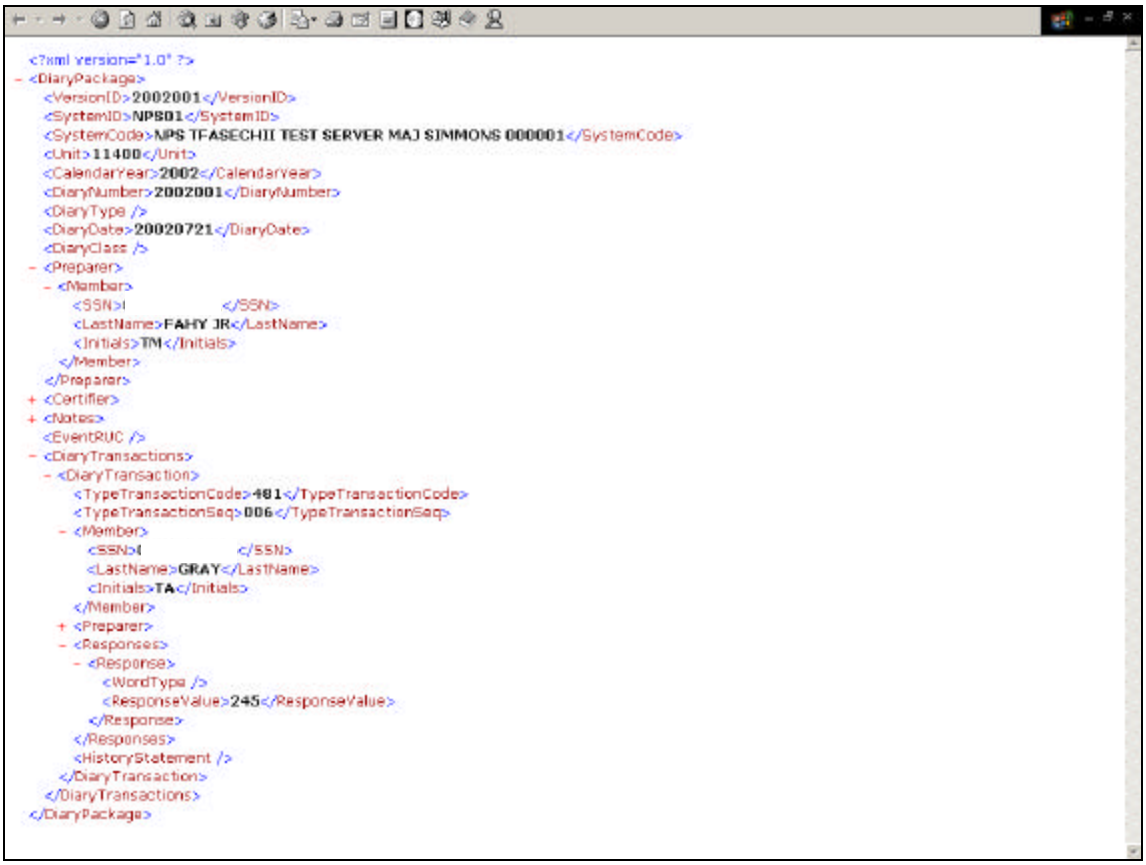


Figure 35. Sample Web Generated XML Unit Diary – PFT

The code that generates the XML unit diary file is depicted in Figure 36. Upon examination, one can see that the code is specifically written for a PFT diary transaction for the data entry of a single PFT for an individual Marine. While this code demonstrates our proof of technical concept for getting web generated data into a format acceptable for MCTFS transactions, it may not be the best solution to the data entry task.

```
‘Create the XML DOM object
Set oDOM = server.CreateObject("Microsoft.XMLDOM")
oDOM.preserveWhiteSpace = True
Set root = oDOM.createElement("DiaryPackage")
oDOM.appendChild root

‘Create nodes of the XML DOM
Set VerID = oDOM.createElement("VersionID")
Set SysID = oDOM.createElement("SystemID")
Set SysCode = oDOM.createElement("SystemCode")
Set RUCUnit = oDOM.createElement("Unit")
Set CalYear = oDOM.createElement("CalendarYear")
Set DNum = oDOM.createElement("DiaryNumber")
Set DType = oDOM.createElement("DiaryType")
Set DDate = oDOM.createElement("DiaryDate")
Set DClass = oDOM.createElement("DiaryClass")
    •
    •
    •

'General Diary Data
VerID.Text = VersionID
SysID.Text = SystemID
SysCode.Text = SystemCode
RUCUnit.Text = Unit
CalYear.Text = CalendarYear
DNum.Text = DiaryNum
DDate.Text = DiaryDate

'Append 1st Level Children
root.appendChild VerID
root.appendChild SysID
root.appendChild SysCode
root.appendChild RUCUnit
root.appendChild CalYear
root.appendChild DNum
root.appendChild DType
root.appendChild DDate
root.appendChild DClass
    •
    •
    •

‘Save the XML DOM as an XML file to a specified directory
fileName = DiaryDate & "PFTDiary" & MarineSSN
Set objPI = oDOM.createProcessingInstruction("xml", "version='1.0'")
oDOM.insertBefore objPI, oDOM.childNodes(0)
dim filePath
filePath = "C:\TFASDiary\" & fileName & ".xml"
oDOM.save filePath
```

Figure 36. Sample Code - XML DOM Object

There are two issues to consider regarding the optimal programming solution to data entry and XML Unit Diary generation. The first issue deals with determining the best method for data entry from the point of view of the TFAS user. When entering data, like PFT Scores, should the user be presented with: (1) one record at time as in the sample shown, (2) all Marines in the unit on one page, or (3) a search capability to locate a specific Marine? This choice of user interface for data entry will affect how the XML Unit Diary file is created programmatically. A single XML Unit Diary file may contain multiple sets of data entry items (e.g. multiple sets of PFT scores). From the global perspective of diary processing on the MCTFS mainframe it would be more efficient for each diary to contain multiple data transactions. The designers of TFAS will need to study this issue and decide on the optimal solution for both the data entry interface for the TFAS user and diary processing efficiency.

The second issue surrounding the optimal programming solution for dynamic XML Unit Diary creation deals with code reuse across the entire application and the task of updating code in the future. If each ASP page in the TFAS web application has its own lengthy code specific to that task, then the code is not particularly modular. Any change to the TTC and SEQ codes, or other MCTFS unit diary data fields and values, would require the programmer to review all web pages that could be affected. The solution to this problem is modularization of the XML Unit Diary file generation task. If a COM object could be developed that is general enough to handle all common diary tasks, then the task of generating XML Unit Diaries could be isolated to one place in the code. It would make programming all of the data entry tasks much more efficient through code reuse and would simplify future updates of the code when the MCTFS programmers make changes to the MCTFS schema for unit diaries. To make this modularization even more efficient, the designers of TFAS could also separate the "data" of TTC and SEQ codes (MCTFS numerical codes representing specific diary transactions) from the code. If a relational database was created that contained the current set of TTC and SEQ codes, the COM object created for XML Unit Diary generation could access the TTCSEQ database at run time and obtain the most current values of TTC and SEQ codes. When MCTFS programmers make changes to the TTC

and SEQ numbers, the database could be updated, and no code would need to be altered in the COM object for XML file generation function or the TFAS ASP pages. This modularization of tasks and data separation would make the TFAS web application highly flexible and resilient to changes in MCTFS systems over time.

The creation of a COM object and a separate database for the TTC and SEQ codes were not implemented in the prototype for several reasons. In order for the COM object to be truly portable across all TFAS data entry pages, it must be general enough to handle all types of diary transactions. An in-depth understanding of all possible unit diary transactions is needed before programming could begin. Only trained administrators (01XX MOS) with years of experience truly understand the great variety of complex diary transactions described in the two inch thick MCO P1080.40C, Marine Corps Total Force System Personnel Reporting Instructions Manual. Such an analysis of diary transactions was beyond the scope of this thesis. The creation of the TTC and SEQ code database was similarly impractical to implement because each specific TTC and SEQ code sequence is tied to a diary transaction. Depending on the type of transaction tied to the TTC and SEQ number, there is a wide variety of "Prompts" and supporting fields that must be described in the database schema. Again a detailed understanding of this unit diary process is required in order to accurately construct the schema of such a database.

The purpose of this chapter was to detail the programming efforts made to construct the TFAS, Echelon II web site prototype. This included a description of our approach to web page authoring, a web site layout, explanation of the login sequence, sample code for a report and data entry task, and the definition of the XML unit diary. This presentation documented one possible implementation of the TFAS, Echelon II web site. The prototype developed for this thesis has demonstrated our goal of "technical proof of concept." The operation of prototype has been presented to several audiences of fellow students and instructors at NPS. The presentations showed a TFAS user logging onto the TFAS, Echelon II web site, viewing several reports, and entering data. An examination of the XML Unit Diary created as a result of the data entry was also conducted. While not all functions described in Chapter 4 were implemented, several examples of each major requirement were demonstrated. Programming will continue on

this prototype after this thesis is published so that it may be used as a working example of TFAS implementation. It will be up to the TFAS program decision makers at MARCORSYSCOM, M&RA (manpower authorities), and TSO programmers at DFAS to determine if this prototype is appropriate as a basis for requirements for an operational system and/or as an initial development template. In the final chapter, a summary of information presented in this thesis and the recommendations for the TFAS program, will be given. In addition, some observations and recommendations will be provided to aid these agencies as they make decisions regarding the future of the TFAS program.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSION AND RECOMMENDATIONS

A. RECOMMENDATIONS

In the course of the research for this thesis, some important aspects of TFAS development have been discovered. These "lessons learned" should be carefully considered as TFAS moves from concept to reality.

Definition of Systems Architecture. A decision regarding detailed systems architecture should be made as soon as possible. Current planning documents are too broad or are silent on detailed systems architecture. In this thesis, two approaches were described: the single national web/database server model and a distributed system of web and database servers located at major bases. TFAS planners must conduct a detailed analysis of which approach is the best solution. Factors such as server load, response time, code maintenance and upgrade, equipment and software costs, facility and manning requirements, web site and database administration procedures, database synchronization, and customer service should be analyzed. This thesis suggests the distributed approach mainly due to shortened client response time and reduced server load. A distributed architecture would also provide for manageable user account administration, flexibility for innovation, speedy troubleshooting, and a higher level of customer service.

Technology Selection. A decision must be made regarding the specific software products to be used in TFAS. Our prototype used all Microsoft products, which provides the benefits of integrated user accounts and system interoperability. Other systems may be more appropriate, however. For example, the Marine Corps uses Oracle products in a number of its systems, and this may be the database of choice for contractual reasons. Whatever software products are selected, it is important to ensure that they are interoperable.

Definition of User Requirements. Define the user functional requirements for each echelon in detailed terms. The functional requirements should be described in enough detail (similar to Echelon II requirements defined in this thesis) so that a programmer can actually use them to develop code. The TFAS working group should be convened and not disbanded until all the requirements at each Echelon and transactions across Echelons have been detailed. The working group should also include some

knowledgeable web and database programmers, as well as several non-administrative Marines that could represent the interests of Echelon I, II and III. Currently, the TFAS working group is manned by administrators who, while very professional and motivated, have a naturally biased view of TFAS functionality at each Echelon.

Empower Lower Echelons. Give as much data entry capability to lowest TFAS Echelon as possible. A central tenet of TFAS is to partition the administrative tasks bottlenecked in the PAC out to the "users" of data – Marine leaders who actually know the Marines who are the subject of the data transaction. In this thesis, several examples, such as PFT score and duty status, were cited. Critics of company data entry can only have one justification for their reasoning: they do no trust the company commander and staff. The answer for the critics is to "trust but verify." Review of the data is appropriate and expected at the battalion level. TFAS functionality can be developed for Echelon III to allow for such oversight.

Morning Reports. Allow company level TFAS users to enter duty status codes. If the MCTFS manpower data actually contained accurate duty status codes on a daily basis, a wealth of service-wide information would be available to leaders at every level. It is only at the company level that leaders really know their Marines and see them on a daily basis. The morning report compiled at the company level is the cornerstone of accountability and the source of accurate manpower status. If we allow the leaders at the company level to use TFAS to enter duty status codes into MCTFS on a daily basis, we will have a uniform morning report process whose aggregate data will form a fairly accurate readiness picture Marine Corps-wide. Other types of aggregate and statistical data could also be formed from the variety of status codes. This type of enterprise-wide daily status information currently eludes us.

Develop a Deployment Plan. Once the system architecture is defined, technology components selected, and Echelon user functional requirements defined, a master software deployment should be drawn up. This deployment plan should be divided into phases similar to the deployment of Marines and equipment in an amphibious operation. When Marines goes to war, the entire force does not arrive in theater on the same day. The force is deployed in pieces according to a master deployment plan. Such is the case with TFAS. First, a TFAS test bed of a few units at a base should be stood up. After an

evaluation period and implementation of recommended changes, TFAS could be rolled out Corps-wide. Additionally, TFAS should be deployed "in-Echelon" over time. With the level of complexity for data transactions increasing with each TFAS Echelon, it may take some time to develop the higher echelons. Whereas, a limited TFAS Echelon II web site could be developed and deployed in the very near future.

Data Entry Methods. The optimal data entry user interface should be determined. The prototype implemented the data entry task by displaying a single Marine's record at a time. For multiple data entry tasks (e.g. multiple PFT scores within a unit), the user had to page through each record in the unit – searching for each Marine who was the subject of the data entry. This interface is one solution. Other options include: 1) displaying all personnel on one web page, and 2) a search capbility to find a specific Marine in the unit. The method of data entry for multiple data transactions will have an impact on Unit Diary processing efficiency and the nature of the programming code supporting the multiple data entry task.

MCTFS Schema Modification. Once a detailed list of functional requirements for all Echelons of TFAS users are defined, there will be data field requirements identified that are currently absent in the MCTFS Central Master File. In this thesis, the defining of functional requirements identified several "missing" data fields for an individual Marine's record in MCTFS. One important set of missing fields were platoon and company codes for each RUC. It was identified that there are six different Reporting Unit Codes (RUC) in the "Individual_Marine" Table of the ODSE, but only one platoon code and one company code. Because a Marine can, at any given time, be assigned to several RUCs, each of these units can overwrite the platoon and company code field values. This situation prevents accurate platoon and company level views of all permanently assigned and temporarily assigned personnel. Since queries for Echelon II will be based on the company and platoon data fields, the MCTFS Central Master File schema must be augmented to include a platoon code and company code for each of the six RUCs if Echelon II functionality is to be successful.

Automate XML Unit Diary Import/Translation. The key to data entry of MCTFS data via a web page in TFAS is the creation an XML file containing the unit diary data. The XML file is translated into the proprietary MCTFS data format using the generic

import utility in UDMIPS. This import utility is a manual process. While this import utility has solved the technical challenge of translating web data to MCTFS unit diary data, it is not the optimal enterprise-wide solution. As TFAS is developed, the quantity of XML Unit Diaries created via the web will grow exponentially. The daily manual translation of these XML files will soon become impractical. From a total systems perspective, the designers of TFAS must develop an automated capability for this translation. Further, it might be more productive to eliminate the middleman (UDMIPS) altogether. Program the MCTFS mainframe so that the "Cycle" could read in the XML diaries directly. The issue of XML Unit Diary translation and loading to the MCTFS main-frame will be a hugely important issue as the quantity of data transactions grows as a result of empowering more users with data entry capability.

Development of a XML Diary COM Object. The creation of the XML Unit Diary in the prototype developed for this thesis used programming code that was specific to one task for each data entry ASP page. While this approach solves the problem, it lacks portability and makes updating software problematic. A COM program should be written that generalizes the XML Unit Diary file creation process across all TFAS data entry web pages. Changes to the MCTFS unit diary schema would require the updating of only one section of code, the COM object, versus hundreds of TFAS web pages.

Development of TTC and SEQ Codes Relational Database. Separate data from the code. Currently, all MCTFS data entry transactions (unit diaries) are defined by an extensive list of TTC and SEQ codes. The TTC and SEQ codes change from time to time as updates are made to the MCTFS. In our prototype, the TTC and SEQ values are "hard-coded." Changes in TTC and SEQ values would require the updating of web page programming code. The solution to this software maintenance and upgrade issue is to separate the data (TTC and SEQ codes) from the programming. A relational database of all the MCTFS TTC and SEQ codes should be created. If such a database existed, the appropriate TTC and SEQ numbers could be read into the COM Object/ASP page at run time. Changes to TTC and SEQ numbers would then not affect any of the TFAS programming.

Development of a Complex Transactions Relational Database. Complex transactions could be implemented into the prototype by creating a relational database or

new tables in the existing backend database. This database would maintain the state of data in a complex transaction between TFAS users over time. At any instant of time during the complex transaction, the data would be tagged with a specific TFAS user's User ID. TFAS users would see they have a pending transaction when they log in to the TFAS web site. As each TFAS user takes action on a multi-step transaction, the state of transaction's data is updated in the transaction data database and tagged with the next user's ID.

TFAS linkage to E/MSS. Functional requirements for TFAS describe the need for a Marine leader to view a Marine's Leave and Earnings Statement (LES). Financial data is sensitive and private and we did not have access to financial data. Currently, individual Marines can log onto a web site run by DFAS and view their LES. This system is called the Employee/Member Self-Service (E/MSS) application. The programmers at TSO/DFAS must work on the technical problem of linking a TFAS web site to the E/MSS application.

B. FURTHER WORK

The TFAS program is still in its infancy and the need to develop and deploy it grows every day. As a result, there is no shortage of research that could be conducted by Naval Postgraduate School students to aid in TFAS development. The following items describe some ideas for further work.

Dynamic Query Capability. The data views or reports developed for the prototype were devised from standard reports available in UDMIPS. While these reports will satisfy the majority of data needs of leaders, the system is static and inflexible. Leaders will have data needs that these predefined reports do not satisfy. As a result, functionality should be developed that allows TFAS users to create ad-hoc queries via the TFAS web site. Incorporating this type of capability into TFAS would ensure customer buy-in to the TFAS application.

Functional Requirements. Develop detailed functional user requirements for all Echelons of TFAS, including the cross-functional, complex transaction tasks.

Security Analysis. This thesis addressed some security issues and incorporated standard web security methods such as Secure Socket Layer and access control through

Windows permissions. However, due to the scope of the entire TFAS program, a thorough security analysis is recommended. Students in the Security track could conduct such an analysis, simulate attacks on the TFAS web site prototype and recommend and/or construct programmatic security measures to incorporate into the TFAS design.

Systems Architecture. A thorough analysis of the most appropriate system architecture for the entire TFAS system is needed. A cost benefit analysis should entail server load, response time, code maintenance and upgrade, equipment and software costs, facility and manning requirements, web site and database administration procedures, database synchronization, and customer service.

XML Unit Diary Generation Function. Programming of a code that generalizes the XML Unit Diary file generation process for all TFAS data transactions is needed. Such a project should also incorporate the development of a relational database for MCTFS TTC and SEQ codes in order to separate the data from programming.

Echelon III (Battalion) Prototype Web Site. Develop TFAS, Echelon III functional requirements and fully program a web site prototype. The scope of the project would be similar to this thesis.

Development of Relational Database for Complex Transactions. Develop a relational database to keep state of data during intermediate steps in a multi-step, multi-user, multi-echelon data transaction. The database would be accessed by the TFAS web site to present the state of data for an administrative request from the requesting Marine to one or more leaders in his/her chain of command.

C. CONCLUSION

The purpose of this thesis was to identify and analyze the requirements and develop a prototype web site for The United States Marine Corps' Total Force Administration System (TFAS), Echelon II – A Web Enabled Database for The Small Unit Leader. The analysis was accomplished by researching the characteristics of the current manpower system, MCTFS, and the conceptual tenets of the TFAS program. This research combined with the author's experience as a small unit leader provided the foundation for the detailed presentation of functional requirements and system architecture for the Echelon II web site. Once the requirements and architecture were

defined, an operational web site prototype for TFAS, Echelon II was developed. Having fulfilled the goal of the thesis, the purpose of this chapter is to present some conclusions, recommendations, and suggestions for further work regarding our analysis and the development and deployment of TFAS, Echelon II.

The goal of the TFAS program is to leverage current and emerging IT technologies to revolutionize our aging manpower data systems and processes. The need for this revolution has grown over the past decade as the current manpower data systems increasingly failed to meet the manpower data needs of Marine leaders. The central effort behind TFAS is to web-enable MCTFS manpower data and to partition data transactions from the single chokepoint of the PAC to leaders at various command levels. While the TFAS concept has been in existence for five years, it has only recently received serious attention. This is mainly due to the manpower cuts in administrative personnel and the consolidation of battalion administrative units to the MSC level over the past two years. These actions caused the current manpower systems to become even more strained in an effort to meet the data needs of Marine leaders. This situation set the stage for this thesis work.

Currently, the TFAS program is still in the concept exploration phase as a procurement program. With the exception of Echelon I (Individual Marine), the functional requirements defined for TFAS pertain to generalized ideas for the entire system. No detailed overall system architecture has been defined. In this thesis we defined detailed system and user functional requirements for the Echelon II. While the scope of these functional requirements dealt only with atomic transactions at the small unit level, they can be used as starting point for overall system integration. Multi-step, multi-echelon transactions were not presented, as they would have required a total systems approach. However, the prototype did demonstrate that the "atomic" transactions for Echelon II could be implemented rather easily. If this limited TFAS, Echelon II web site were deployed, the TFAS developers could begin to ease some of the administrative burden on the overworked and undermanned PACs. The fielding of this slice of the TFAS program would also provide a platform by which the TFAS developers could gain some detailed and practical insight into the technical challenges and new business rules inherent to the TFAS program as a whole.

The information presented in this thesis should be viewed through the lens of the limitations and scope of the prototype. The prototype was developed with the singular focus of TFAS, Echelon II atomic data transactions. The final TFAS web site must be able to handle multi-step, multi-echelon transactions. This requirement could alter the proposed system architecture presented in this thesis. However, the system architecture presented in this thesis should be scalable to an enterprise-wide solution. Also, in order to develop a working prototype, specific software technologies had to be selected. The assumption of a Windows NT/2000 network environment and the selection of the IIS-5 Web Server and SQL Server 2000 database forced certain design decisions in the construction of the prototype. While the prototype represents "a" solution among many possible commercial-off-the-shelf (COTS) technologies, at present, it is the "only" solution. Lastly, the programming used to develop the prototype were the efforts of a single, relatively inexperienced individual. Due to the magnitude and impact of the TFAS program, a team of experienced web programmers should develop any deployed TFAS web site. This statement, however, should not cause the reader to discount the potential worth of the prototype. TFAS planners have already spent hundreds of thousands of dollars on TFAS "studies." These studies have yielded insightful macro views of the issues involved with TFAS development, but have not set forth detailed system architecture or user functional requirements, much less any operational prototypes. The prototype developed for this thesis was virtually cost-free and should demonstrate to TFAS planners that TFAS development need not proceed at snail's pace nor cost a fortune. While the prototype only addresses a narrowly focused slice of the functional domain of TFAS, it could serve as a template for the development of each Echelon. The distributed system architecture for TFAS and integrated Microsoft components outlined in this thesis, as demonstrated by the prototype, can easily be scaled to the total TFAS solution. It is hoped that this thesis work will provide some detailed insight to TFAS planners at MARCORSYSCOM, M&RA, and TSO/DFAS so that the TFAS program may progress beyond conceptual planning.

APPENDIX A GLOSSARY

AFADBD – Armed Forces Active Duty Base Date. The date that appeared on the first contractual agreement (enlistment document) for a member of the armed forces.

API – Application Programming Interface. An API is the specific method prescribed by a computer operating system or by a software application program by which a programmer writing an application program can make requests of the operating system or another application.

ASP – Active Server Page. Microsoft's web page scripting technology. Essentially, an ASP file is are a hybrid of static HTML markup code and programmatic code based on a derivation of the Visual Basic programming language called Visual Basic Script (VB Script). The use of ASP "web pages" allows the web page author to dynamically create a web page based on the inputs to the VB Script. Ultimately, the VB Script in an ASP file outputs HTML code back to the requesting web client.

ASVAB – Armed Services Vocational Aptitude Battery. A written examination that tests general educational level and cognitive skills. Administered to all entrants of the armed forces and is used to qualify individuals for various job skills, enlistment programs, and commissioning programs.

BAS – Basic Allowance Subsistence. A non-taxable pay entitlement for subsistence (food). Mostly associated with Marine Officers. The equivalent entitlement for enlisted Marines is COMRATS. See COMRATS.

BIR – Basic Individual Record. Standardized UDMIPS report detailing the basic data fields in MCTFS for a Marine. See Table 8, Chapter 4 for data fields.

BST – Basic Skills Test. A written and oral examination administered to junior enlisted Marines annual to test their proficiency of basic military skills and knowledge.

BTR – Basic Training Record. Standardized UDMIPS report detailing current training data fields for a Marine. See Table 8, Chapter 4 for data fields.

CMF – Central Master File. The CMF is the data portion of the MCTFS mainframe. This data is in a flat file format and contains all of the data for over 500,000 active duty, reserve, and retired Marines.

COM - Component Object Model (COM) is Microsoft's framework for developing and supporting program component objects. It is aimed at providing similar capabilities to those defined in the Common Object Request Broker Architecture (CORBA), a framework for the interoperation of distributed objects in a network that is supported by other major companies in the computer industry.

COMRATS – Commuted Rations. A non-taxable pay entitlement for subsistence (food) for enlisted personnel. There are several different monthly COMRATS rates depending on Marine's martial status and housing status.

CONAD – Consolidated Administration. Also known as "Consolidated Admin". An administrative unit staffed by specially trained manpower data administrators, serving multiple battalion-sized units. All administrative functions and manpower and pay data transactions are conducted at the CONAD. Synonymous with PAC.

CUDDB – Commanding Officer's Unit Diary Database. Local copy of the MCTFS Central Master File. This database contains a replicated copy of the manpower data resident in MCTFS. Local admin units connect to this database to read manpower data into their UDMIPS desktop application. The CUDDB is refreshed (updated) with data updates from MCTFS after unit diaries (data transactions) are processed on the MCTFS mainframe.

DBMS – Database Management System. A DBMS is a program that lets one or more computer users create and access data in a database. The DBMS manages user requests (and requests from other programs) so that users and other programs are free from having to understand where the data is physically located on storage media and, in a multi-user system, who else may also be accessing the data. In handling user requests, the DBMS ensures the integrity (that is, making sure it continues to be accessible and is consistently organized as intended) and security (making sure only those with access privileges can access the data) of the data. The most typical DBMS is a relational database management system (RDBMS). A standard user and program interface is the Structured Query Language (SQL). See SQL.

DCTB – Date Current Tour Began. The date on which a Marine was assigned and reported for duty at their present duty station.

DECC - Defense Enterprise Computing Center. The agency and physical location of the mainframe computer that houses the single source of Marine Corps manpower data, MCTFS.

DFAS – Defense Finance and Accounting Service. The DOD agency responsible for maintaining and operating all of the uniformed services financial and pay data systems. Because pay and benefits for a service member is integrally tied to manpower data, many of the services manpower data systems agencies are collocated or are integrated with DFAS. DFAS is located in Kansas City, MO and is staff by civilians and uniformed personnel.

DOR – Date of Rank. The date on which a service member was promoted to their present rank.

FSSG – Force Service Support Group. The logistical component of a MEF. A large unit consisting of multiple battalions with specialized logistical support capabilities.

HOR – Home of Record. The address and state within the United States that a Marine claims as his/her legal, permanent address. Usually it is the state in which the Marine resided when he/she entered active duty service.

IIS-5 – Internet Information Server-5. Microsoft's current web server that comes standard on all Windows 2000 family operating systems. Capacity (number of connected users) and features vary by type of Windows 2000 operating system and by number of licenses purchased for the web server.

LES – Leave and Earnings Statement. A monthly pay and benefits statement for an individual Marine. It is the equivalent to a "pay stub" in civilian industry.

M&RA – Manpower and Reserve Affairs. The Headquarters Marine Corps agency responsible for all aspects of personnel management and issues for the Marine Corps. They develop and implement all personnel administration policy. They are the co-owner of MCTFS and related manpower data systems and are functional sponsor of all administrative personnel (01XX MOSs).

MDA - Milestone Decision Authority. The individual who holds a billet (job) within a Department of Defense or uniformed service acquisition agency that has the authority to make major decisions regarding the status of a procurement program. These decisions are typically associated with signature/approval authority of key procurement documents and transitions of the program from one acquisition phase to another.

MARCORSYSCOM – Marine Corps Systems Command. The Marine Corps' agency responsible for all acquisition programs including information systems. MARCORSYSCOM is responsible for the development of the TFAS program. Located in Quantico, VA.

MCC – Monitored Command Code. A 3-digit numeric code that uniquely identifies a command, activity, or individual billet to which assignment of personnel is controlled by Headquarters Marine Corps. MMCs are typically associated with major subordinate commands (e.g. Marine Division, Wing, and FSSG).

MCO – Marine Corps Order. Acronym for official Marine Corps policy documents.

MCTFS - Marine Corps Total Force System. The single, integrated, personnel and pay system supporting both active and reserve components of the Marine Corps. Encompasses all of the personnel and pay data as well as the software programs that process the data for over 500,000 active, reserve and retired Marines.

MEF – Marine Expeditionary Force. A large warfighting organization in the Marine Corps. Can contain 20,000 to 40,000 Marines and Sailors. Four major components: Command Element, Ground Combat Element (Division), Service Support Element (FSSG), and Air Combat Element or ACE (Wing). There are three active duty and one reserve MEFs in the Marine Corps.

MISSO – Marine Information Systems Support Office. Regional manpower data systems support agencies. There are about a half dozen MISSO organizations; typically one per major Marine Corps installation. Provides support, training, and software maintenance to local administrative units.

MOL – Marine OnLine. The web site at www.mol.usmc.mil. Individual Marines set up a personal account at this web site where they can access their personal manpower data. The site almost all read-only, but the TFAS developers are attempting to morph this web site into TFAS, Echelon I and add the "write" functions where individual Marines can actually make some limited MCTFS data transactions.

MOS – Military Occupational Specialty. A 4-digit numerical code that represents a job skill in the armed forces. Individuals may have several MOSs with one being designated as the primary job skill and the others designated as additional MOSs. See PMOS.

MSC – Major Subordinate Command. Typically associated with a unit that has a MCC designation and is commanded by a general officer. In everyday the term MSC refers to the Marine Division, Wing, and FSSG.

NOK – Next of Kin. The person designated by a service member as being the relative or person to notify in the event the service member is injured, missing in action or killed while on active duty.

NMCI – Navy-Marine Corps Intranet. The effort by the Department of the Navy, which includes the U.S. Navy and U. S. Marine Corps, to integrate all it's information systems. The NMCI effort not only includes information systems policies but a multi-billion dollar contract over ten years with the information technology firm EDS to outsource all hardware and software procurement as well as many garrison based IT management jobs and functions.

ODBC – Open Database Connectivity. ODBC is the standard database access method developed by Microsoft Corporation. The goal of ODBC is to make it possible to access any data from any application, regardless of which database management system (DBMS) is handling the data. ODBC manages this by inserting a middle layer, called a database driver, between an application and the DBMS. The purpose of this layer is to translate the application's data queries into commands that the DBMS understands. For this to work, both the application and the DBMS must be ODBC-compliant -- that is, the application must be capable of issuing ODBC commands and the DBMS must be capable of responding to them.

ODSE – Operational Data Store Enterprise. The ODSE is a commercial relational database, which contains a copy of the manpower data resident in the MCTFS CMF. It is updated nightly with the data changes after the MCTFS cycle and as a whole twice yearly when software changes are made to MCTFS. It is presently an Oracle 8i Enterprise database.

OLDS - On-Line Diary System. A MCTFS application subsystem used to access the manpower data in MCTFS. Characterized by an "Atari" black and white screen command line prompts and function keys interface. All data entry must be inputted manually by looking up the correct codes, as there are no drop-down menus or lists to select data. Currently being phased out of use for the windows like application called UDMIPS.

PAC – Personnel Administration Center. A Marine Corps unit that is staff by specially trained administrative personnel and serves several battalion/squadron sized units. Also known as CONAD (Consolidated Administration). It is at the PAC or CONAD, that all manpower data transactions (interface with MCTFS) occur.

PEBD – Pay Entry Base Date. The date on which a member of the armed forces first entered active duty service.

PersO – Personnel Officer. Warrant Officers in the Marine Corps (MOS 0110). Specially trained in manpower data and pay processes and data systems. The term PersO usually referred to the officer in charge of the administrative unit within a battalion. As a special battalion staff officer, he/she was responsible directly to the commanding officer of the battalion or squadron for the daily operations of the unit's admin shop and the maintenance of the unit's manpower data. With the advent of the PACs, this role no longer exists and multiple personnel officers work in the PACs and are responsible to the commanding General of the MSC. Also known as Admin Officer.

PFT – Physical Fitness Test. A physical fitness test administered to all active duty Marines semi-annually.

PMOS – Primary Military Occupational Specialty. The primary job skill of an individual in the armed forces. See MOS.

RUC – Reporting Unit Code. A 5 digit numerical code that uniquely identifies a unit that has reporting responsibility and/or authority. Reporting requirements are usually identified with battalion/squadron sized units or independent commands that submit unit diaries to update MCTFS data. RUCs are also used to identify echelons of command, which may not submit unit diaries (e.g. division, regiment, wing, etc.)

SQL – Structured Query Language. SQL is a standard interactive and programming language for getting information from and updating a database. Although SQL is both an ANSI and an ISO standard, many database products support SQL with proprietary extensions to the standard language. Queries take the form of a command language that lets you select, insert, update, and delete data in a relational database

TFAS - Total Force Administrative System. TFAS is the technical implementation of the Marine Corps' upgrade of human resource system payroll and personnel administration services concept. This concept and technical architecture seeks to reorganize current business processes; organization structure; implement new policy and procedures; and to align information technology (IT) assets around a data-centric environment. The ability to communicate, share, and manipulate large amounts of data across a distributed operational environment is the key tenet behind this concept.

TSO – Technology Services Organization. An agency within DFAS whose is responsible for the technical programming and maintenance of a variety of Marine Corps manpower data systems including the MCTFS mainframe, UDMIPS, and the ODSE.

TRECON – Transaction Reconciliation. The process whereby local admin units download and extract data from the MCTFS Central Master File. This extract file contains the latest updated manpower data in MCTFS after a MCTFS cycle. The extract is then used to update the CUDDB, the local MCTFS manpower data source used by the UDMIPS MCTFS application sub-system.

TTC – Type Transaction Code. A 3-digit numeric code that represents specific unit dairy data transactions.

UDMIPS – Unit Diary/Marine Integrated Personnel System. UDMIPS is a heavy client, windows driven application used to view data and make data changes to MCTFS data. Using UDMIPS, the administrative clerk (Unit Diary Clerk) can view and print preformatted reports on personnel in the unit. For data updates, UDMIPS is used to enter the data updates in a windows-like interface with menus. UDMIPS then converts the human readable data into a file consisting of a series of alphanumeric codes formatted for transmission to the MCTFS mainframe to be processed so as to update the Central Master File.

UD - Unit Diary. The process of making data updates to MCTFS or, rather, the set of data representing the data update. The set of data is ultimately transformed into a file that consists of alpha-numeric codes and data that is in a proprietary format that is acceptable as data input for processing the data update on the MCTFS mainframe.

XML – Extensible Markup Language. XML, a formal recommendation from the World Wide Web Consortium (W3C), is similar to the language of today's Web pages, the Hypertext Markup Language (HTML). Both XML and HTML contain markup symbols to describe the contents of a page or file. The difference being that HTML describes the content of a Web page (mainly text and graphic images) only in terms of how it is to be displayed and interacted with. For example, the letter "p" placed within markup tags starts a new paragraph in a HTML web page. XML on the other hand describes the content in terms of what data is being described. XML is a language for writing a language to organize data and to describe the meaning of the content of the data.

LIST OF REFERENCES

1. MCO P1080.40C Marine Corps Total Force System Personnel Reporting Instructions Manual (MCTFSPRIM), 7 Jun 2002.

2. Polach, J. and Young, K., Unpublished Paper: Marine Corps Total Force System: The Workhorse for personnel and Pay Throughout the Marine Corps, 2000.

3 Gaskins, G., Unpublished Paper: Total Force Administration System (TFAS) – Bringing The Marine Corps Administration System "On-Line" with Today and Tomorrow, 2000.

4 MCO 1080.33E Manpower Information Systems Support Office Procedures (MISSO Manual), 20 Mar 2000.

5. Peterson, J., USMC Information Paper: The Changing Face of Marine Corps Administration, USMC M&RA, 1999.

6. Total Force Administrative System (TFAS) Statement of Work (SOW), 11 Oct 2000.

7. Gaskins, G., USMC Information Paper: Total Force Administration System (TFAS) – Bringing The Marine Corps Administration System "On-Line" with Today and Tomorrow, 2001.

8. Total Force Administrative System (TFAS) Baseline Requirements Matrix, 3 Oct 01.

9. MARCORSYSCOM TFAS Program Brief, Feb 2000.

10. MCO 5720.76 Standardization of Publicly Accessible Web Pages, 14 Sep 01.

11. United States Marine Corps World Wide Web Style Guide, 1 Aug 01.

12. Microsoft Windows 2000 Server, Microsoft Press, 2000.

13. Reeves, S. and Reeves, K., IIS 5 On Site, The Coriolis Group, 2001.

14. Shapiro, J. R., SQL Server 2000 The Complete Reference, The McGraw-Hill Companies, 2001.

15. Thurrott, P, et al., Visual InterDev 6 Unleashed, Sams Publishing, 1999.

16. Mohr, S., et al., XML Application Development with MSXML 4.0, Wrox Press, 2002.

17. Deardorff, G., Unpublished Paper: DiaryPackage Document, 2001.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Fort Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California

4. Director, Training and Education, MCCDC, Code C46
   Quantico, Virginia

5. Director, Marine Corps Research Center, MCCDC, Code C40RC
   Quantico, Virginia

6. Marine Corps Tactical Systems Support Activity (Attn: Operations Officer)
   Camp Pendleton, California

7. Director, Information Systems & Infrastructure Product Group (PG10)
   Marine Corps Systems Command (MARCORSYSCOM)
   Quantico, Virginia

8. Director, Technology Services Organization (TSO) – Kansas City
   Defense Finance and Accounting Service (DFAS)
   Kansas City, Missouri

9 Mr. David L. Ferris, Jr.
   Product Group Director, Information Systems and Infrastructure
   MARCORSYSCOM
   Quantico, Virginia

10. Lieutenant Colonel Scott R. Adams
    Program Manager, Combat Support Information Systems
    MARCORSYSCOM
    Quantico, Virginia

11. Mr. James D. Kenkel
    Deputy Branch Head, MIP
    Manpower & Reserve Affairs
    Quantico, Virginia

12. Dr. Daniel R. Dolk
Information Systems Department
Naval Postgraduate School
Monterey, California

13. Dr. Thomas Otani
Computer Science Department
Naval Postgraduate School
Monterey, California

14. Major Steven A. Simmons, USMC
Montgomery, Alabama

15. Dr. Martha Poole Simmons
Montgomery, Alabama